



UNIVERSIDADE DE SÃO PAULO  
ESCOLA DE ENGENHARIA DE SÃO CARLOS  
Departamento de Engenharia Elétrica e de Computação

# Projeto Final Grupo 10 - SEL0632

## Redes Neurais Artificiais

Fulvio Favilla Filho  
Guilherme Lopes Matias  
João Antônio Evangelista Garcia

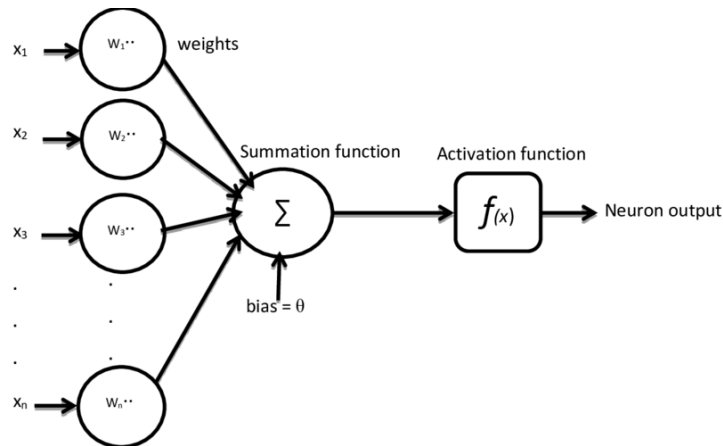
Docente: Dr. Maximilian Luppe

São Carlos  
25/05/2021

# 1 Introdução

## 1.1 Neurônio Artificial

Neurônios artificiais recebem este nome, pois são inspiradas no funcionamento dos neurônios do cérebro humano. Um neurônio na computação recebe sinais e os processam para gerar uma determinada saída. Além disso, uma característica importante é sua capacidade de aprender e melhorar o desempenho após uma etapa de treinamento.



**Figura 1:** Neurônio Artificial [1]

O neurônio artificial é formado por:

- Sinais de entrada ( $x_1, x_2, \dots, x_n$ )
- Pesos sinápticos ( $w_1, w_2, \dots, w_n$ )
- Combinador linear ( $\Sigma$ )
- Limiar de ativação ( $\theta$ )
- Sinal de saída ( $y$ )
- Função de ativação ( $f$ )

As expressões a seguir apresentam o resultado produzido pelo neurônio:

$$u = \sum_{i=1}^n w_i \cdot x_i - \theta$$
$$y = f(u)$$

O funcionamento do neurônio consiste em receber os sinais de entrada e multiplicar cada um por um determinado peso, baseado na importância para a saída. Depois o somador realiza uma soma ponderada e, comparando com um certo limiar de ativação, o neurônio aciona ou não o sinal de saída.

O aprendizado do neurônio é feito por meio de um algoritmo seguindo uma lista de regras pré-definidas alterando tanto o limiar de ativação quanto os pesos sinápticos até obter os resultados desejados.

## 1.2 Operações em ponto fixo (*Q-format*)

As operações em ponto fixo permitem a representação aritmética de valores reais por meio da manipulação de valores inteiros. Esta forma de representação é essencial em aplicações que necessitam de alto desempenho ou em *hardwares* de baixo custo que não possuem unidade de ponto flutuante.

A notação *Q-format* é a mais comum para representar um número em ponto fixo. A denotação  $Q_{m.n}$  indica em  $m$  a quantidade de bits da parte inteira, enquanto  $n$  é a quantidade de bits da parte fracionária e, pelo número poder ser positivo ou negativo, também é reservado um bit para o sinal. Por exemplo, o formato  $Q_{2.8}$  necessita de 11 bits para representar um número real: 2 bits para a parte inteira, 8 bits para a parte fracionária e 1 bit para o sinal.

A conversão de números reais binários para o formato  $Q_{m.n}$  é descrita pela seguinte equação:

$$X = x \times 2^n \quad (1)$$

A equação (1) expressa a relação entre um número real  $x$ , na base 2, e o seu equivalente  $X$  no formato  $Q_{m.n}$ .

## 2 Função de ativação

Como mostrado na figura 1, o último passo na execução do neurônio é a passagem pela função de ativação ( $g$ ), pois é o que analisa se ele deve ser ativado, ou seja, esta função avalia se a informação recebida é relevante ou não. Para cumprir essa função existem diversas formas de funções com diferentes necessidades de processamento e também com precisão variada.

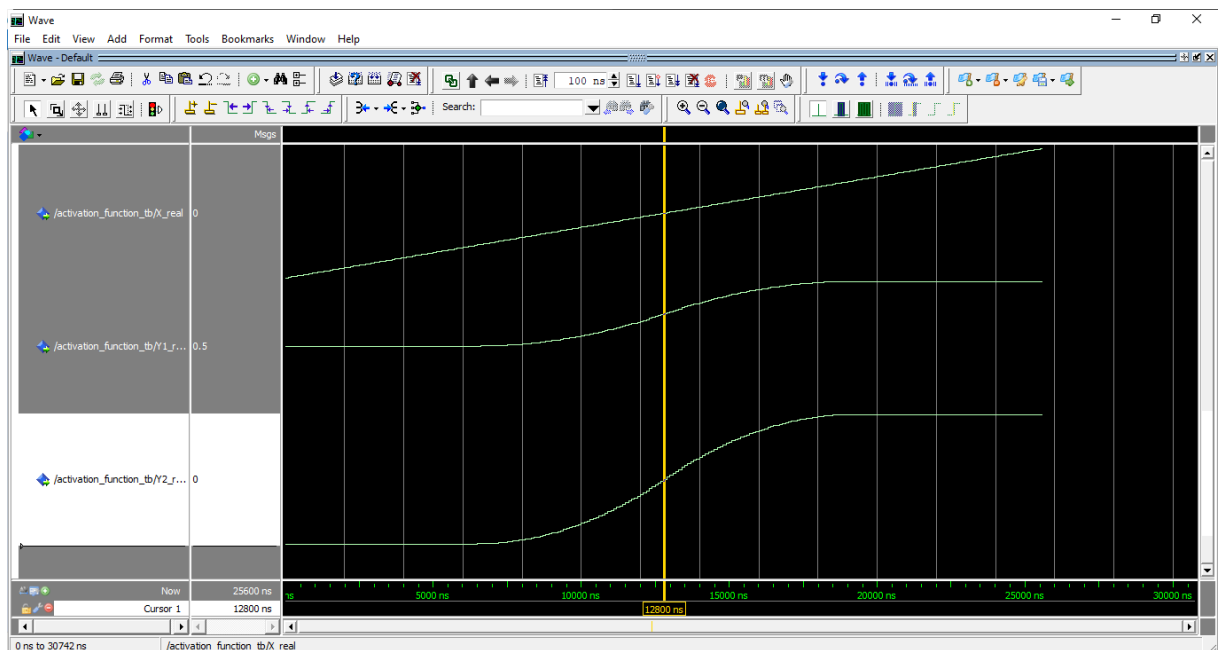
Para o projeto do neurônio foram utilizadas como funções de ativação a função sigmoide e a função tangencial hiperbólica. Em sua implementação e teste foram considerados na entrada da função sigmoide valores no formato  $Q_{3.11}$ , dessa forma os valores do potencial de ativação ( $u$ ) seriam de -8 até 8, para os testes foram utilizadas duas funções de ativação, a primeira que possui valores de saída ( $y$ ) entre 0 e 1, e a segunda com valores de saída ( $y$ ) entre -1 e 1. As funções foram implementadas de forma que a saída vá do valor mínimo ao máximo no intervalo de valores de entrada  $-4 < u < 4$ , assim para

valores de entrada menores que -4 a saída é mínima (0 para a primeira função e -1 para a segunda função) e para valores maiores que 4 a saída é máxima, valendo 1 para as duas funções.

### 3 Resultado

O material utilizado para o trabalho, incluindo a descrição em VHDL do neurônio, está disponibilizado no GitHub a seguir: [https://github.com/fulvio-f/ann\\_mlp](https://github.com/fulvio-f/ann_mlp)

Seguem as formas de onda obtidas no resultado final:



**Figura 2:** Formas de onda obtidas no ModelSim

### Referências

- [1] “Figure 3.2 The structure of the artificial neuron.”, ResearchGate. [https://www.researchgate.net/figure/The-structure-of-the-artificial-neuron\\_fig2\\_328733599](https://www.researchgate.net/figure/The-structure-of-the-artificial-neuron_fig2_328733599) (acessado ago. 13, 2021).
- [2] I. Tsmots, O. Skorokhoda, e V. Rabyk, “Hardware Implementation of Sigmoid Activation Functions using FPGA”, in 2019 IEEE 15th International Conference on the Experience of Designing and Application of CAD Systems (CADSM), fev. 2019, p. 34–38. doi: 10.1109/CADSM.2019.8779253.