# Relatório: Inferência de Rede Neural em RISC-V

Renilson C. de Luna Jr.

24 de junho de 2025

## 1 Introdução

Este relatório descreve a implementação de um processo de inferência de rede neural utilizando arquitetura RISC-V de 32 bits. O objetivo é executar uma rede previamente treinada com pesos quantizados e otimizados, usando operações compatíveis com instruções básicas da ISA RISC-V.

O sistema recebe os dados de entrada, executa as camadas da rede (como convoluções, ativações e normalizações) e gera a saída inferida com uso mínimo de instruções e memória. Este trabalho foca em eficiência, portabilidade e clareza do código, destacando os desafios encontrados e as soluções propostas.

## 2 Módulo de Entrada e Parsing

A leitura dos dados de entrada é uma etapa crucial para o funcionamento da rede neural. O parser é responsável por interpretar os dados brutos e convertê-los em um formato que possa ser utilizado pela rede. A leitura dos pesos da rede foi feita aproveitando o fato de que a quantidade de vetores de uma camada i corresponde ao número de neurônios da camada, e cada vetor contém o número de pesos correspondente ao número de neurônios da camada anterior, segundo o exemplo:

$4, 10, 20, 3 \rightarrow$ 4 camadas:

- Entrada: 4 neurônios
- Oculta 1: 10 neurônios
- Oculta 2: 20 neurônios
- Saída: 3 neurônios

Logo, as camadas lN possuem matriz no formato:

- l1: [10][4]
- l2: [20][10]
- l3: [3][20]

Com os tamanhos dos vetores em mãos, e dado que cada inteiro é separado por vírgula, o parser lê os dados de entrada seguindo a seguinte regra:

1. Lê o número de camadas da rede;
2. Lê o número de neurônios da camada de entrada;
3. Interpreta os tamanhos dos vetores de cada camada, armazenando-os em um vetor ln;
4. Lê os pesos de cada camada, pulando a vírgula entre os números;

Assim, é possível construir as matrizes de pesos de cada camada, armazenando-as em vetores de inteiros com tamanho o suficiente para conter todos os pesos em um vetor unidimensional inicializado com 0's, considerando a maior quantidade de neuronios observada (50):

```
.data
  # Vetores de pesos para cada camada
  arquitetura: .space 20        # 4 valores de Entrada
  l1: .space 200                # 50 * 4 valores de Oculta 1
  l2: .space 2500               # 50 * 50 valores de Oculta 2
  l3: .space 2500               # 50 * 50 valores de Oculta 3
  l4: .space 2500               # 50 * 50 valores de Oculta 4
  ln: .space 20                 # Vetor de tamanhos das camadas
  saida: .space 3
```