

## **Lista #8**

**Curso: Ciência da Computação**

**Disciplina: Inteligência Artificial**

**Prof<sup>a</sup>. Cristiane Neri Nobre**

---

## **Redes Neurais com Backpropagation**

---

**Objetivo:** Implementar, a partir do zero (sem bibliotecas como Keras, PyTorch ou Scikit-learn), o algoritmo de **backpropagation** para treinar uma rede neural simples que reconheça:

- 1) O problema do XOR

Mostre a equação de ajuste dos pesos durante o processo de treinamento

- 2) Dígitos binários de um **display de 7 segmentos**.

Para este problema, imagine um display digital (como o de um relógio ou calculadora), formado por **7 LEDs** (segmentos) que podem estar ligados ou desligados para formar os dígitos de 0 a 9.

Vamos simplificar para reconhecer **os dígitos de 0 a 9** com base na ativação dos 7 segmentos.

---

## **Tabela de entrada e saída esperada**

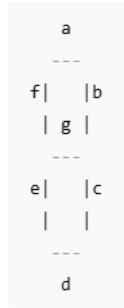
---

Cada entrada é um vetor de 7 posições (segmentos a a g).

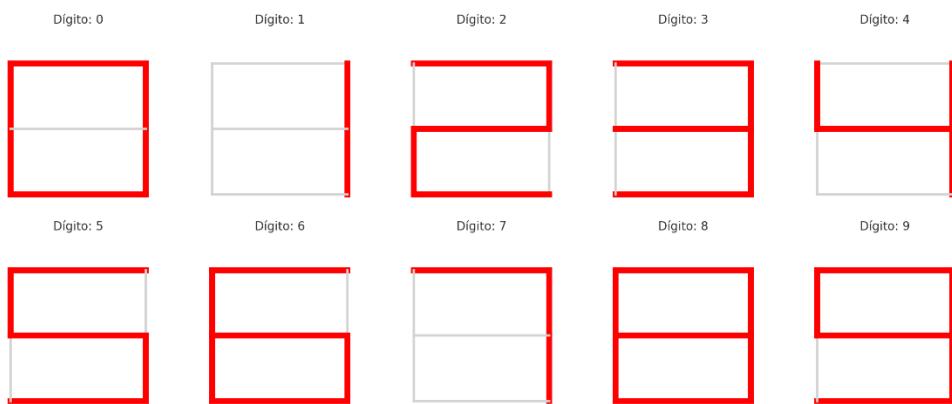
**Exemplo:**

Dígito	Segmentos Ativos (binário)	One-hot Output
0	[1, 1, 1, 1, 1, 1, 0]	[1, 0, 0, 0, 0, 0, 0, 0, 0]
1	[0, 1, 1, 0, 0, 0, 0]	[0, 1, 0, 0, 0, 0, 0, 0, 0]
2	[1, 1, 0, 1, 1, 0, 1]	[0, 0, 1, 0, 0, 0, 0, 0, 0]
3	[1, 1, 1, 1, 0, 0, 1]	[0, 0, 0, 1, 0, 0, 0, 0, 0]
4	[0, 1, 1, 0, 0, 1, 1]	[0, 0, 0, 0, 1, 0, 0, 0, 0]
5	[1, 0, 1, 1, 0, 1, 1]	[0, 0, 0, 0, 0, 1, 0, 0, 0]
6	[1, 0, 1, 1, 1, 1, 1]	[0, 0, 0, 0, 0, 0, 1, 0, 0]
7	[1, 1, 1, 0, 0, 0, 0]	[0, 0, 0, 0, 0, 0, 0, 1, 0]
8	[1, 1, 1, 1, 1, 1, 1]	[0, 0, 0, 0, 0, 0, 0, 1, 0]
9	[1, 1, 1, 1, 0, 1, 1]	[0, 0, 0, 0, 0, 0, 0, 0, 1]

## Display de 7 Segmentos: Esquema de referência



Veja os prints abaixo:



## Estrutura da rede

- 7 neurônios na camada de entrada (um para cada segmento).
- 5 neurônios na camada oculta.
- 4 neurônios na camada de saída (usando codificação one-hot).

**Use a função sigmoide:**

---

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

E investigue outras funções de ativação (RELU, softmax, etc)

**Calcular o erro (loss):**

---

Função de erro quadrático médio (MSE):

$$E = \frac{1}{n} \sum_i (y_i - \hat{y}_i)^2$$

Para os testes, além dos dígitos originais, adicionar ruído às entradas (simulando falha de algum segmento) e verificar a robustez do modelo.

## Entregáveis

---

1. Link para o código desenvolvido
2. O relatório PDF com todas as explicações, prints de entrada e saída da rede neural, mostrando os acertos e erros do modelo, para cada um dos problemas acima.