

Projeto 2 - Modelo não supervisionado

Guilherme Pereira Campos RA:163787
Lucas Oliveira Nery de Araújo RA:158882
Universidade Federal de São Paulo

I. RESUMO

Este projeto tem como objetivo aplicar redes neurais não supervisionadas, utilizando o modelo SOM (*Self-Organizing Map*), para explorar a detecção de padrões em dois *datasets* distintos. O primeiro *dataset* envolve imagens médicas de raios-X de tórax para classificação de pneumonia, enquanto o segundo contém dígitos manuscritos representados por matrizes de pixels.

O modelo foi utilizado para identificar *clusters*, *outliers* e avaliar a homogeneidade dos agrupamentos em ambos os conjuntos de dados. Adicionalmente, variações nos parâmetros da rede, como tamanho do *grid*, número de neurônios, taxa de aprendizado e sigma, foram exploradas para analisar seu impacto na qualidade dos padrões detectados. Este projeto destaca a aplicabilidade do modelo SOM em cenários de aprendizado não supervisionado.

II. REDE SOM

A. Aprendizado Não Supervisionado

As redes SOM (*Self-Organizing Maps*) utilizam aprendizado não supervisionado, identificando padrões nos dados sem a necessidade de rótulos ou supervisão. Este tipo de aprendizado explora a redundância presente nos dados para organizar informações em um espaço mais simples.

B. Aprendizado Competitivo

O aprendizado competitivo, fundamental para o funcionamento das redes SOM, permite que os neurônios "compitam" para se ativar com base no padrão de entrada. Apenas o neurônio vencedor e seus vizinhos têm seus pesos ajustados. Isso resulta em uma organização dos dados e na identificação de padrões.

C. Mapa Auto-Organizável (SOM)

As redes SOM são modelos neurais inspirados na organização topológica do córtex cerebral, como os córtices visual, auditivo e motor. Utilizando aprendizado competitivo, esses modelos organizam os neurônios em um *grid* (geralmente bidimensional), onde unidades próximas no mapa respondem a estímulos semelhantes. O treinamento de uma rede SOM envolve três etapas principais:

- **Competição:** Os neurônios competem para determinar qual será ativado com base no padrão de entrada.
- **Cooperação:** Os neurônios vizinhos ao vencedor ajustam seus pesos.
- **Adaptação Sináptica:** Os pesos são ajustados para refletir os padrões de entrada.

1) *Fase de Competição:* Na fase de competição, cada neurônio da rede calcula um valor com base em uma função discriminante, determinando qual deles é mais próximo do padrão de entrada. O neurônio com o melhor resultado é chamado de *Best Matching Unit* (BMU) e declarado o vencedor. O cálculo do BMU é realizado com base no vetor de entrada \mathbf{x} e nos pesos dos neurônios \mathbf{W}_j , conforme a equação:

$$i(\mathbf{x}) = \arg \min \|\mathbf{x} - \mathbf{W}_j\|$$

2) *Fase de Cooperação:* Após determinar o BMU, define-se uma vizinhança topológica em torno do neurônio vencedor. Essa vizinhança é definida pela proximidade dos neurônios no *grid*, independentemente de seus pesos. A função de vizinhança é dada por:

$$h_{j,i}(\mathbf{x}) = \exp \left(-\frac{d_{j,i}^2}{2\sigma^2} \right)$$

onde $d_{j,i}$ é a distância entre os neurônios j e i no mapa. O parâmetro σ decai com o tempo, seguindo:

$$\sigma(t) = \sigma_0 \exp \left(-\frac{t}{\tau_\sigma} \right)$$

onde, η_0 é a taxa de aprendizado inicial e τ_η é o tempo de decaimento.

3) *Fase de Adaptação:* Os pesos dos neurônios na vizinhança do BMU são ajustados para aproximar-se do padrão de entrada. O ajuste é realizado pela fórmula:

$$\mathbf{W}_j(t+1) = \mathbf{W}_j(t) + \eta(t) \cdot h_{j,i}(\mathbf{x})(t) \cdot (\mathbf{x} - \mathbf{W}_j)$$

onde:

- $\eta(t)$ é a taxa de aprendizado, que decai exponencialmente ao longo do tempo:

$$\eta(t) = \eta_0 \exp \left(-\frac{t}{\tau_\eta} \right)$$

- $h_{j,i}(\mathbf{x})(t)$ é a função de vizinhança centrada no BMU.

D. Treinamento

O treinamento de uma rede SOM ocorre em duas fases principais:

- 1) **Organização Topológica:** Nesta etapa inicial, o mapa organiza os neurônios para representar a topologia dos dados de entrada.
- 2) **Ajuste Fino:** Os pesos sinápticos são ajustados de forma mais precisa para especializar os neurônios em padrões específicos.

Durante o treinamento, os parâmetros η (taxa de aprendizado) e σ (função de vizinhança) são ajustados dinamicamente. O decaimento gradual desses parâmetros é essencial para garantir a convergência da rede e a formação de uma ordenação topológica eficiente no mapa.

III. IMPLEMENTAÇÃO

A. Datasets Utilizados

O primeiro dataset utilizado é o *Chest X-ray Images (Pneumonia)*, disponível no *Kaggle* (<https://www.kaggle.com/datasets/paultimothymooney/chest-xray-pneumonia>). Este dataset contém 5856 imagens de raios-X de tórax, divididas em duas classes: pneumonia (bacteriana e viral) e sem pneumonia (normal), com as imagens rotuladas manualmente.

O segundo dataset é o *Digits Dataset*, que contém 1797 imagens de dígitos manuscritos, representados por matrizes de 8x8 pixels, totalizando 64 características por imagem. As imagens já estão prontas para uso, com os respectivos dígitos (0 a 9) rotulados.

B. Preparação dos Dados

A preparação dos dados foi realizada apenas para o primeiro dataset (*Chest X-ray Images*) e seguiu as seguintes etapas:

- **Leitura das Imagens:** As imagens foram organizadas em pastas com as classes "PNEUMONIA" e "NORMAL". A biblioteca `os` foi utilizada para navegar entre as pastas e acessar os arquivos.
- **Redimensionamento:** As imagens foram redimensionadas para um tamanho fixo utilizando a biblioteca `OpenCV (cv2)`.
- **Transformação:** As imagens foram achatadas em vetores para o processamento na rede, utilizando a função `img.flatten()`.
- **Codificação de Rótulos:** As classes foram convertidas para valores binários para representar as categorias.

A biblioteca `cv2` foi usada para carregar as imagens, convertê-las para escala de cinza com `cv2.imread()` e redimensioná-las com `cv2.resize()`. A biblioteca `os` foi utilizada para listar arquivos (`os.listdir()`) e construir caminhos de forma compatível entre sistemas operacionais (`os.path.join()`).

C. Rede SOM

A rede SOM foi implementada utilizando a biblioteca *MiniSom* (<https://github.com/JustGlowing/minisom>), uma implementação simples do algoritmo SOM em Python. A rede SOM foi treinada e testada com os dados dos dois datasets, visando identificar padrões e agrupar as imagens com base em características semelhantes.

IV. MODELOS

A. Primeiro modelo

O primeiro modelo SOM foi configurado com um grid 4x4 (16 neurônios), taxa de aprendizado de 0.5% e sigma de 0.5, sendo treinado por 10 épocas. Apesar da configuração simples,

o modelo alcançou uma acurácia satisfatória, demonstrando eficiência na identificação de padrões nos dados.

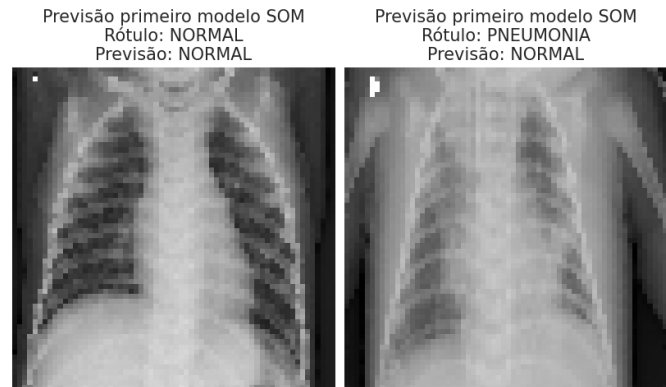


Fig. 1. Previsão primeiro modelo SOM

1) *Proporção de amostras por classe em cada neurônio:* A proporção de amostras por classe em cada neurônio associa visualmente as classes representadas. Isso é feito utilizando gráficos de setores para cada neurônio, permitindo compreender como o SOM organiza e distribui as classes no mapa.

Proporção de Amostras por Classe em Cada Neurônio Primeiro Modelo

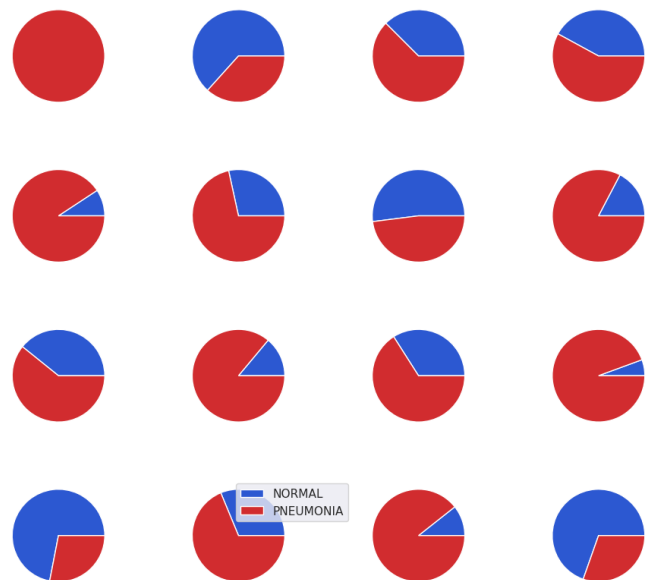


Fig. 2. Proporção de Amostras por Classe em Cada Neurônio Primeiro Modelo

2) *Erro de quantização:* O erro de quantização mede a média das distâncias entre cada dado e o neurônio vencedor no SOM, indicando o quão bem os dados foram representados pelo modelo. Valores menores refletem maior qualidade no mapeamento dos padrões.

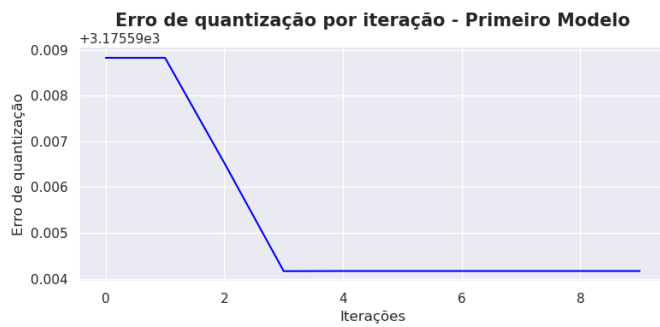


Fig. 3. Erro de quantização por iteração - Primeiro Modelo

B. Segundo modelo

O segundo modelo SOM utilizou um grid de 8x8 (64 neurônios), configurado com uma taxa de aprendizado de 0.01 e sigma de 1.5. Após 200 iterações de treinamento, o modelo mostrou-se eficiente na identificação de padrões. Para validação, duas amostras aleatórias foram selecionadas, com suas previsões apresentadas graficamente, além das classes dominantes de cada neurônio e o erro de quantização.

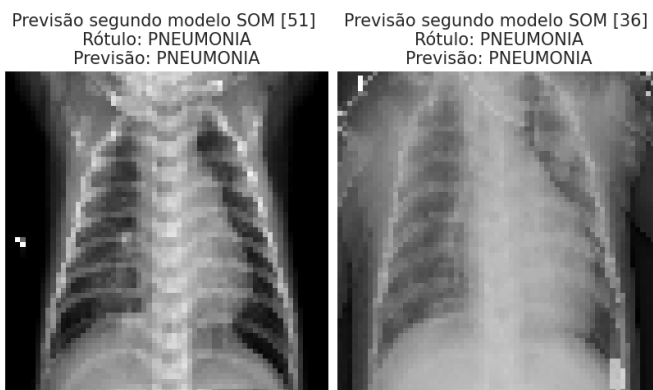


Fig. 4. Previsão segundo modelo SOM



Fig. 6. Erro de quantização por iteração - Segundo Modelo

Os modelos subsequentes foram treinados utilizando o dataset `load_digits` da biblioteca `scikit-learn`.

C. Terceiro modelo

O terceiro modelo SOM utilizou um grid de 5x5 (25 neurônios), com taxa de aprendizado de 0.05 e sigma igual a

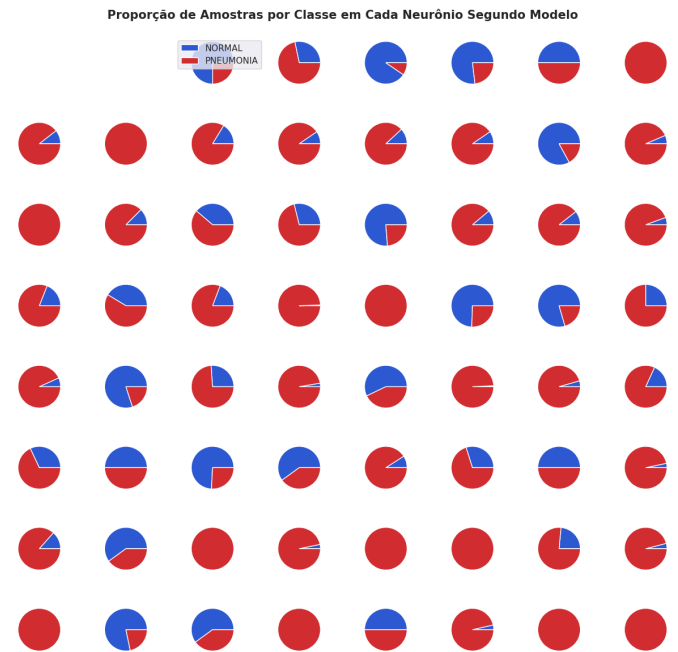


Fig. 5. Proporção de Amostras por Classe em Cada Neurônio Segundo Modelo

1. Após 50 iterações, apresentou acurácia significativa. Foram plotados um exemplo correto e um incorreto, além das classes dominantes de cada neurônio e o erro de quantização.

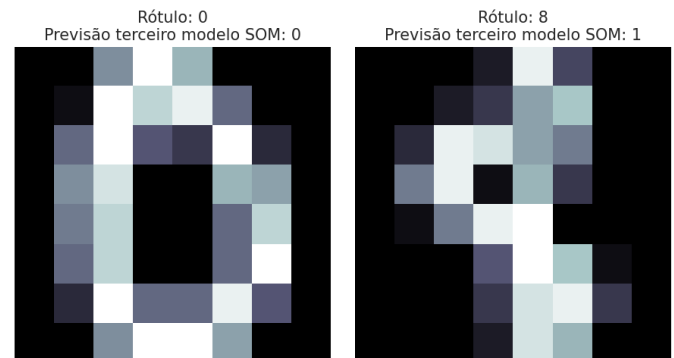


Fig. 7. Previsão terceiro modelo SOM

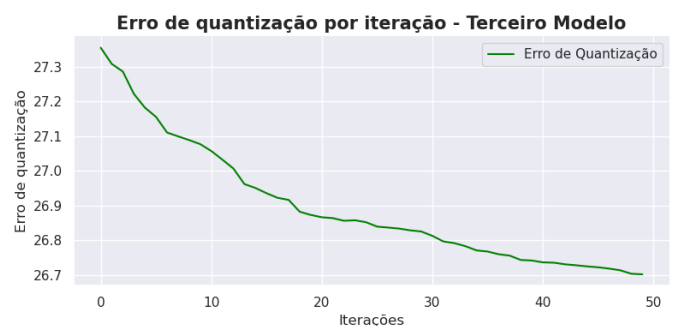


Fig. 8. Erro de quantização por iteração - Terceiro Modelo

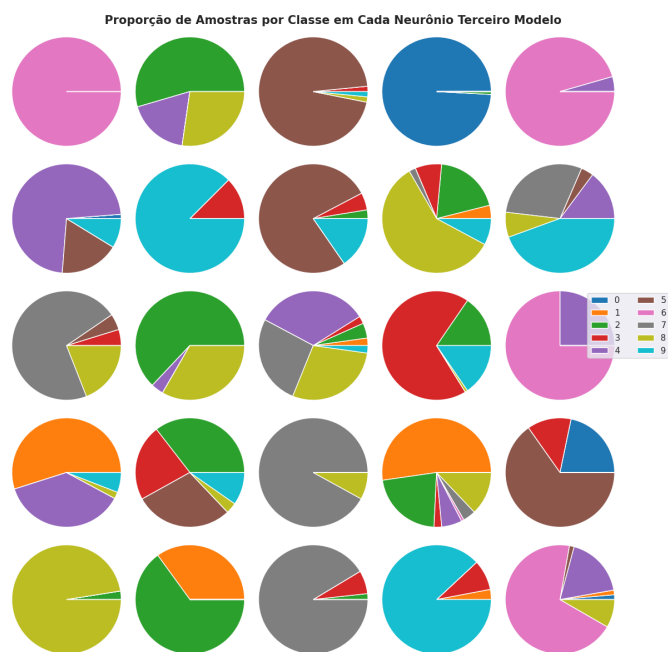
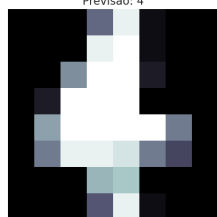


Fig. 9. Proporção de Amostras por Classe em Cada Neurônio Terceiro Modelo

D. Quarto modelo

O quarto modelo SOM foi configurado com um grid de 10x10 (100 neurônios), taxa de aprendizado de 0,5 e sigma de 5. Após 200 iterações de treinamento, o modelo demonstrou boa capacidade de organização dos dados. Duas amostras aleatórias foram selecionadas, cujas previsões foram apresentadas graficamente, além das classes dominantes de cada neurônio e o erro de quantização.

Previsão quarto modelo SOM, exemplo aleatório: 126
Rótulo verdadeiro: 4
Previsão: 4



Previsão quarto modelo SOM, exemplo aleatório: 372
Rótulo verdadeiro: 8
Previsão: 8

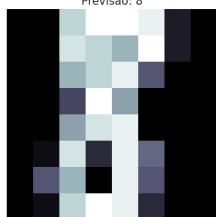


Fig. 10. Previsão quarto modelo SOM

V. CONCLUSÃO

Em conclusão, este projeto demonstrou a eficácia do modelo SOM (Self-Organizing Map) na detecção de padrões e na organização de dados em dois datasets distintos: imagens de raios-X de tórax para diagnóstico de pneumonia e dígitos manuscritos. A aplicação da rede SOM permitiu identificar clusters, outliers e avaliar a homogeneidade dos agrupamentos, além de explorar a influência dos parâmetros de rede, como tamanho do grid, taxa de aprendizado e sigma. A análise revelou que, mesmo com configurações simples, o modelo



Fig. 11. 'Erro de quantização por iteração - Quarto Modelo'



Fig. 12. Proporção de Amostras por Classe em Cada Neurônio Quarto Modelo

SOM conseguiu classificar eficientemente os dados, destacando sua aplicabilidade em aprendizado não supervisionado.