Mapas Auto-Organizáveis e Autoencoders

Elias C. Rodrigues[1], **Leonardo N. Matos**[2]

[1]Departamento de Computação (DCOMP) – Universidade Federal de Sergipe (UFS)

[2]Departamento de Computação (DCOMP) – Universidade Federal de Sergipe (UFS)

elias.rodrigues@dcomp.ufs.br, leonardo@dcomp.ufs.br

Abstract. *This paper presents an idea that combines SOM (Self-Organizing Map) and convolutional autoencoders to create a hierarchical and interpretable model. The proposed method replaces each neuron in the SOM grid with an autoencoder. The SOM organizes the grid topologically, creating neighborhood relationships among the autoencoders. Each autoencoder serves as a feature extractor, as its encodings store the main features of an input in a latent representation. In a separate training phase, with the SOM training already completed, the latent layers generated by the autoencoders are used in Multilayer Perceptron (MLP) networks for image classification tasks.*

Resumo. *Este trabalho apresenta uma ideia que combina SOM (Self-Organizing Map) e autoencoders convolucionais para criar um modelo hierárquico e interpretável. O método proposto substitui cada neurônio da grade do SOM por um autoencoder. O SOM organiza a grade topologicamente, criando relações de vizinhança entre os autoencoders. Cada autoencoder funciona como um extrator de características, já que suas codificações armazenam as principais características de uma entrada em uma representação latente. Em um treinamento separado, com o treinamento do SOM já realizado, as camadas latentes geradas pelos autoencoders são utilizadas em redes Perceptron Multicamadas (MLP) para tarefas de classificação de imagens.*

1. Introdução

Nos anos 80, Teuvo Kohonen desenvolveu os mapas auto-organizáveis de características [Kohonen 1982], habitualmente chamados de *Self-Organizing Map* (SOM) ou Mapa Auto-Organizável. SOM é um paradigma de redes neurais de aprendizado não supervisionado que, por meio de uma topologia fixa, mapeia os dados de entrada e busca por semelhanças e padrões. Isso torna o modelo particularmente útil para visualização e interpretação de conjuntos de dados complexos. Os neurônios do SOM ficam dispostos em uma grade, sendo a grade bidimensional mais utilizada por facilitar a visualização e interpretação do resultado.

Outra rede neural de aprendizado não supervisionado é o *Autoencoder*. Um *autoencoder* é composto por dois processos que são conhecidos como *encoder* (codificador) e *decoder* (decodificador). Na etapa de codificação, o *autoencoder* codifica a entrada de modo a gerar uma representação mais compacta, comumente conhecida como representação latente. Em seguida, essa representação latente é passada para o decodificador, que por sua vez tenta reconstruir a entrada a partir dessa representação compactada. A Figura 1 exemplifica o funcionamento de um *autoencoder* [Bank et al. 2023].

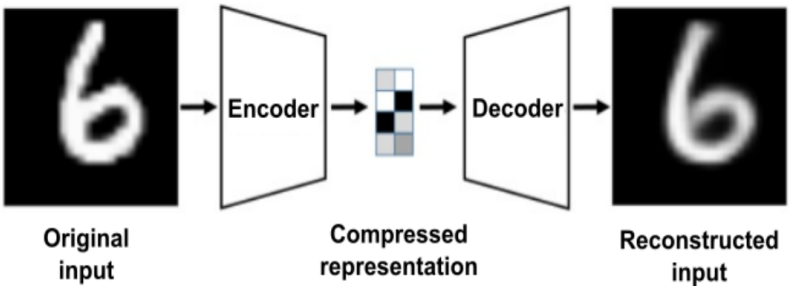

Figura 1. Exemplo do funcionamento do *autoencoder*. A imagem de entrada é comprimida, gerando uma representação latente (*compressed representation*) que é posteriormente decodificada, gerando uma reconstrução da entrada. Fonte: Bank et al. 2023.

Neste trabalho, é proposta uma arquitetura híbrida: serão combinados SOM e *Autoencoder* Convolucional de modo que haja a criação de grupos hierárquicos de *autoencoders* a fim de que uma interpretação seja possível. Cada célula da grade SOM é composta por um autoencoder especializado. Além disso, cada *autoencoder* serve como um extrator de características e sua camada latente será utilizada no treinamento de classificação com a rede MLP acoplada em cada autoencoder da grade. Para os experimentos, serão utilizadas imagens do conjunto de dados MNIST [LeCun et al. 1998].

As principais contribuições deste trabalho incluem: (i) a proposição de um modelo hierárquico e interpretável baseado na integração entre SOM e autoencoders convolucionais; (ii) uma abordagem de classificação descentralizada, com redes MLP especializadas por região da grade; e (iii) uma análise qualitativa e quantitativa da organização topológica e do desempenho do modelo em um cenário com poucos dados de treinamento.

2. Fundamentação teórica

2.1. SOM

O treinamento do SOM consiste em encontrar um neurônio vencedor, também conhecido como BMU (*Best Matching Unit*), em relação a uma amostra de entrada e fazer esse neurônio se aproximar dessa amostra, assim como seus vizinhos da grade. O neurônio vencedor é encontrado calculando uma distância euclidiana entre cada neurônio da grade e a amostra alvo, o neurônio com menor distância será o vencedor. Os neurônios se aproximarão da amostra alvo com uma intensidade que depende diretamente da função de topologia, também conhecida como função de vizinhança. A função de vizinhança utilizada nos experimentos é conhecida como *Gaussian Bell*, segundo a seguinte fórmula:

$$h(i,k,t) = e^{-\frac{||g_i - g_i||}{2 \cdot \sigma(t)^2}} \tag{1}$$

onde g_i e g_k representam a posição do neurônio vencedor e de outro neurônio na grade, respectivamente. A operação $||g_i - g_k||$ é a distância euclidiana entre os dois neurônios na grade e $\sigma(t)$ serve como um raio de vizinhança que decresce com o passar do treinamento, diminuindo a intensidade da função e fazendo com que a adaptação dos neurônios aos dados de entrada seja mais suave [Kriesel 2007]. A Figura 2 exemplifica uma distância euclidiana entre os neurônios i e k, sendo que foi considerada a distância entre cada par de neurônios na horizontal e vertical é equivalente a 1.

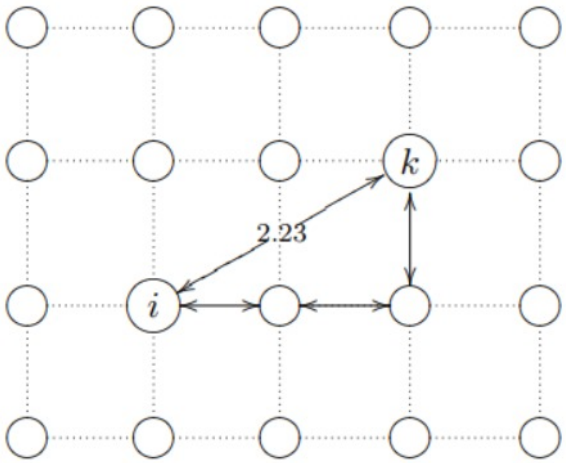

Figura 2. Exemplo de distância euclidiana entre neurônios em uma grade bidimensional com topologia retangular. Fonte: Kriesel, 2007.

A aproximação de cada neurônio para uma amostra alvo de um conjunto de entrada durante o treinamento segue a seguinte regra:

$$c_k(t+1) = c_k(t) + \eta(t) \cdot h(i,k,t) \cdot (p - c_k) \tag{2}$$

onde $c_k(t)$ é o estado atual do neurônio, $\eta(t)$ é a taxa de aprendizado, $h(i,k,t)$ é a função de topologia e $(p - c_k)$ é a diferença entre a amostra alvo e o neurônio. A taxa de aprendizado $\eta(t)$ está em função do tempo porque ela irá decrescer com o decorrer do treinamento [Kriesel 2007].

2.2. Autoencoder convolucional

Um *autoencoder* baseado em Convolutional Neural Network (CNN) utiliza camadas convolucionais e *pooling* na etapa de codificação, para reduzir a dimensionalidade da entrada e filtrar as características mais importantes na camada latente. Já a parte de decodificação utiliza camadas de convolução transposta, de modo que a dimensionalidade volte ao estado inicial e a imagem seja reconstruída [Géron 2021]. CNNs são amplamente conhecidas por sua eficácia quando se trata de extração de características para classificação de imagens [Jogin et al. 2018].

Embora um *autoencoder* seja uma rede neural de aprendizado não supervisionado, é possível utilizá-lo em uma rede neural com camadas totalmente conectadas de aprendizado supervisionado para classificação. Nesse caso, o *autoencoder* servirá como um extrator de características, utilizando o codificador. Um *autoencoder* codifica uma entrada em uma representação latente de dimensão menor, guardando as principais características da entrada nessa representação compactada. A camada latente e um rótulo posteriormente são passados para o treinamento de classificação da rede neural totalmente conectada [Bank et al. 2023].

2.3. Integrações entre SOM e autoencoders

Diversos trabalhos recentes têm explorado a combinação de SOMs com técnicas de extração de características. [Khacef et al. 2020] propuseram o uso de autoencoders como

pré-processadores para SOMs, melhorando a qualidade da organização topológica. Já [Huijben et al. 2023] introduziram o SOM-CPC, uma integração entre SOM e aprendizado contrastivo, com foco em sequências temporais.

Diferentemente desses trabalhos, a proposta deste artigo integra diretamente os autoencoders à estrutura do SOM: cada célula da grade é um autoencoder convolucional especializado, treinado com base em sua vizinhança topológica. Essa abordagem possibilita uma organização hierárquica e interpretável.

3. Metodologia

3.1. Arquitetura geral do modelo

O modelo SOM utilizado é composto por uma grade bidimensional de tamanho 7×7, na qual cada célula corresponde a um exemplo de um *autoencoder* convolucional. Cada *autoencoder* foi implementado com quatro camadas convolucionais 2D na etapa de codificação, com filtros 3×3, ativação *Rectified Linear Unit* (ReLU) e operações de *Max-Pooling* 2×2 após a segunda e a terceira camadas, para reduzir a dimensionalidade. Na etapa de decodificação, são utilizadas quatro camadas de convolução transposta, também com filtros 3×3 e ativação ReLU, intercaladas com operações de *upsampling* para restaurar a resolução original, sendo a última camada seguida de uma ativação *Sigmoid* para limitar a saída entre 0 e 1.

Associada a cada *autoencoder*, há também uma rede MLP que recebe a representação latente, com dimensão $32 \times 3 \times 3$, e é composta por três camadas totalmente conectadas: a primeira com 128 neurônios, a segunda com 64 neurônios (ambas com ativação ReLU), e a última camada com 10 neurônios correspondentes às classes de saída.

3.2. Treinamento

O modelo foi treinado utilizando o conjunto de dados MNIST. Na etapa de treinamento não supervisionado do SOM, foram utilizadas 100 imagens. Para o treinamento supervisionado das redes MLP, foram utilizadas outras 500 imagens, distintas daquelas utilizadas na etapa anterior.

3.2.1. Treinamento não supervisionado

O treinamento segue uma adaptação da abordagem tradicional do SOM, com as seguintes etapas:

1. Cada autoencoder da grade é inicializado com pesos aleatórios e mantém como atributo uma imagem de referência, inicialmente composta por pixels aleatórios.
2. Para cada imagem do conjunto de treinamento, calcula-se a distância euclidiana entre a imagem de entrada e a imagem reconstruída por cada autoencoder da grade.
3. O autoencoder cuja reconstrução mais se aproxima da imagem de entrada é definido como a BMU.
4. A BMU é atualizada por meio de *backpropagation*, com uma única época de treino com o otimizador Adam e função de perda *Mean Squared Error* (MSE). A imagem de treino é utilizada como alvo de reconstrução.

5. Os autoencoders vizinhos da BMU na grade têm seus pesos ajustados para se aproximarem dos pesos da BMU. A regra de atualização segue a equação clássica do SOM (2) e função de vizinhança *Gaussian Bell* (1).
6. Após a atualização dos pesos, cada autoencoder tenta reconstruir novamente a imagem de entrada.

Esse processo é repetido para todas as amostras do conjunto de treinamento ao longo de 1000 épocas, permitindo que a grade de autoencoders se organize topologicamente de acordo com as similaridades entre as imagens. Os valores iniciais de taxa de aprendizado e raio de vizinhança são, respectivamente, $0{,}2$ e $2{,}0$.

3.2.2. Treinamento supervisionado

Após o treinamento não supervisionado do SOM com autoencoders, realiza-se a etapa de classificação supervisionada, utilizando redes MLP associadas a cada autoencoder convolucional. O processo segue as seguintes etapas:

1. Para cada imagem do conjunto de treinamento, identifica-se a BMU na grade já treinada.
2. A imagem é codificada pela BMU, gerando uma representação latente.
3. Essa representação latente é utilizada como entrada da MLP associada à BMU, enquanto o rótulo da imagem é utilizado como alvo para o treinamento supervisionado.
4. A MLP da BMU é atualizada via *backpropagation*, utilizando 10 épocas de treino com o otimizador Adam e funcão de perda *Cross Entropy Loss*, enquanto os vizinhos na grade atualizam suas MLPs para se aproximarem dos pesos da MLP da BMU, seguindo a regra do SOM (2) e função de vizinhança *Gaussian Bell* (1).

Essa estratégia permite que regiões topologicamente próximas na grade aprendam a classificar imagens semelhantes, promovendo consistência na organização espacial e melhorando a generalização do modelo. Foram utilizadas 1000 épocas no treinamento, com valores iniciais de taxa de aprendizado e raio de vizinhança iguais a $0{,}2$ e $2{,}0$, respectivamente.

4. Resultados e discussões

4.1. Resultado do treinamento não supervisionado

A Figura 3 apresenta a grade do SOM após o treinamento não supervisionado. Cada célula da grade exibe a imagem que representa a especialização do respectivo *autoencoder*. Observa-se que o SOM organizou os *autoencoders* de modo que aqueles com especializações semelhantes ficassem posicionados em regiões próximas, preservando a coerência topológica. Algumas imagens de mesma classe ficaram distantes por possuírem características distintas, como espessura, inclinação etc. Além disso, algumas classes apareceram poucas vezes, como é o caso do autoencoder que representa o número 4. O tamanho da grade tem ligação direta com esse problema. Consequentemente, o resultado do treinamento supervisionado é afetado, pois há pouca diversidade dentro de uma mesma classe para encontrar a BMU mais adequada.

O modelo utilizado apresenta alta complexidade computacional, de modo que utilizar mais imagens e uma grade maior levaria várias horas de treinamento, mas com grande potencial de melhorar os resultados.

Figura 3. Resultado da grade após o treinamento não supervisionado.

4.2. Resultado do treinamento supervisionado

A métrica utilizada para analisar o desempenho da classificação foi a acurácia. Para verificar o resultado, foram utilizadas 10 mil imagens do conjunto de teste da base MNIST. A acurácia obtida foi de 74,81%. O resultado está abaixo de modelos contemporâneos que chegam a alcançar mais de 99,8% de acurácia como em [Byerly et al. 2021]. Porém, o resultado obtido é promissor, considerando o número baixo de imagens utilizadas no treinamento.

5. Conclusão

O modelo proposto se mostrou promissor ao alcançar uma acurácia de 74,81% na base MNIST, mesmo com uma grade bidimensional pequena e um número reduzido de imagens para treinamento. Este trabalho contribui com uma proposta de modelo hierárquico e interpretável sob diferentes perspectivas. A principal fonte de interpretabilidade advém da estrutura do SOM, que organiza os autoencoders em uma grade topológica. Cada célula da grade apresenta uma reconstrução visual típica do padrão que aquele autoencoder aprendeu, permitindo uma inspeção direta dos tipos de entrada que ativam determinada região.

Para trabalhos futuros, será necessário otimizar o modelo para viabilizar o treinamento com um conjunto de dados maior e uma grade bidimensional de maior dimensão. Além disso, a aplicação de técnicas de regularização e data augmentation também poderá melhorar a acurácia e a generalização do modelo. Por fim, com um modelo mais maduro, pretende-se realizar experimentos em conjuntos de dados de maior complexidade.

6. Agradecimentos

Este trabalho foi desenvolvido com apoio do Programa Institucional de Bolsas de Iniciação Científica (PIBIC), da Universidade Federal de Sergipe (UFS), sob a

coordenação da COPES.

Referências

Bank, D., Koenigstein, N., and Giryes, R. (2023). Autoencoders. In Rokach, L., Maimon, O., and Shmueli, E., editors, *Machine Learning for Data Science Handbook*. Springer.

Byerly, A., Kalganova, T., and Dear, I. (2021). No routing needed between capsules. *Neurocomputing*, 463:545–553.

Géron, A. (2021). Aprendizado de representação e aprendizado gerativo com autoencoders e gans. In *Mãos à Obra: Aprendizado de Máquina com Scikit-Learn, Keras & TensorFlow*, chapter 17, pages 437–446. Alta Books, 2nd edition.

Huijben, I., Mettes, P., and Snoek, C. G. M. (2023). Som-cpc: Unsupervised contrastive learning with self-organizing maps. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Early Access.

Jogin, M. et al. (2018). Feature extraction using convolution neural networks (cnn) and deep learning. In *2018 3rd IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT)*, pages 2319–2323. IEEE.

Khacef, A., Hébrail, G., and Gallinari, P. (2020). Improving self-organizing maps with unsupervised feature extraction. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE.

Kohonen, T. (1982). Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43(1):59–69.

Kriesel, D. (2007). Self-organizing feature maps. In *A Brief Introduction to Neural Networks*, chapter 10, pages 147–164. Disponível em: http://www.dkriesel.com.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324.