

Otimização de Hiperparâmetros de Autoencoder para Classificação de Imagens via Estratégia Evolutiva e Otimização por Enxame de Partículas*

Dylan Faria Robson
dylan.robson@edu.ufes.br

27 de julho de 2025

1 Introdução

O desempenho de redes neurais profundas, como os autoencoders (AE), é intrinsecamente ligado à escolha de seus hiperparâmetros. A configuração manual da arquitetura da rede — como o número de camadas, neurônios e a taxa de aprendizado — é uma tarefa complexa, iterativa e que raramente converge para uma solução ótima. Neste contexto, meta-heurísticas bioinspiradas surgem como uma abordagem robusta para automatizar e otimizar essa busca em espaços de alta dimensionalidade. (GILANIE et al., 2025).

Este trabalho propõe uma análise comparativa entre duas meta-heurísticas para a otimização de hiperparâmetros de um autoencoder aplicado à classificação de imagens do dataset MNIST. Foram implementadas a *Estratégia Evolutiva* (ES em inglês), conhecida por sua convergência estável, e a *Otimização por Enxame de Partículas* (PSO em inglês), que se destaca pela sua capacidade de exploração do espaço de busca. O objetivo é determinar qual abordagem produz o modelo com melhor poder de generalização, avaliando o *trade-off* entre o erro de reconstrução do autoencoder e a acurácia do classificador final.

2 Metodologia

A metodologia foi implementada em Python, com o suporte das bibliotecas `scikit-learn`, `numpy` e `deap`. Para garantir a reprodutibilidade dos experimentos, uma semente aleatória foi fixada no início do processo.

2.1 Pré-processamento de Dados

Foram utilizadas 15.000 amostras do dataset MNIST. Os dados foram divididos em conjuntos de treino (80%) e teste (20%). Subsequentemente, 10% do conjunto de treino foi

*O código-fonte desenvolvido para este trabalho está disponível no GitHub: <https://github.com/MrRobson9/NC-Atividade-2>

reservado como um conjunto de validação interna, utilizado exclusivamente para o cálculo da função de aptidão dos algoritmos evolutivos.

Os valores de intensidade dos pixels, originalmente no intervalo $x \in \{0, \dots, 255\}$, passaram por uma normalização min-max para o intervalo $[0, 1]$, conforme a equação:

$$x_{\text{norm}} = \frac{x}{255}$$

Esta etapa é fundamental para estabilizar e acelerar o treinamento da rede neural.

2.2 Arquitetura e Representação do Indivíduo

A arquitetura do autoencoder foi projetada para ser flexível, com a otimização definindo sua profundidade e largura. Cada solução potencial (indivíduo) é codificada por um vetor de valores reais I , que define os hiperparâmetros da rede:

$$I = [n_1, n_2, n_{\text{lat}}, \alpha]$$

onde $n_1 \in [1, 256]$, $n_2 \in [1, 128]$ e $n_{\text{lat}} \in [1, 64]$ representam o número de neurônios na primeira, segunda e terceira (latente) camadas do encoder, respectivamente. A taxa de aprendizado α é otimizada no intervalo $[1 \times 10^{-4}, 5 \times 10^{-3}]$.

O autoencoder é treinado por 5 épocas com o otimizador Adam. Após o treinamento, seu decoder (que espelha a arquitetura do encoder) é descartado, e as características extraídas da camada latente são usadas para treinar um classificador `MLPClassifier` dedicado (camada única, treinado por 60 épocas).

2.3 Função de Aptidão

A qualidade de cada indivíduo é medida pelo Erro Quadrático Médio (MSE em inglês) da reconstrução no conjunto de validação interna. Para um lote de N amostras, onde $\mathbf{x}^{(i)}$ é a imagem original e $\hat{\mathbf{x}}^{(i)}$ é a imagem reconstruída pelo autoencoder, o MSE é dado por:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \hat{\mathbf{x}}^{(i)}\|_2^2$$

Como os algoritmos evolutivos foram configurados para maximizar a aptidão, a função de fitness foi definida como $f(I) = -\text{MSE}_{\text{val}}(I)$. Minimizar o MSE, portanto, equivale a maximizar a função de aptidão.

2.4 Configuração dos Algoritmos

Os parâmetros de configuração para a ES e o PSO foram definidos para garantir um orçamento computacional similar. A Tabela 1 resume os principais parâmetros.

Tabela 1: Parâmetros de configuração para ES e PSO.

Estratégia Evolutiva (ES)		Otimização por Enxame de Partículas (PSO)	
Pais (μ)	8	Partículas	56
Descendentes (λ)	56	Iterações	30
Gerações	30	Inércia (ω)	0.7
Prob. de Mutação (indpb)	0.3	Fator Cognitivo (ϕ_p)	1.5
Avaliações Totais	1.688	Fator Social (ϕ_g)	1.5
		Avaliações Totais	1.736

3 Resultados e Análise

3.1 Análise de Convergência

A Figura 1 ilustra a evolução do melhor valor de fitness (MSE na validação) para cada algoritmo ao longo das iterações.

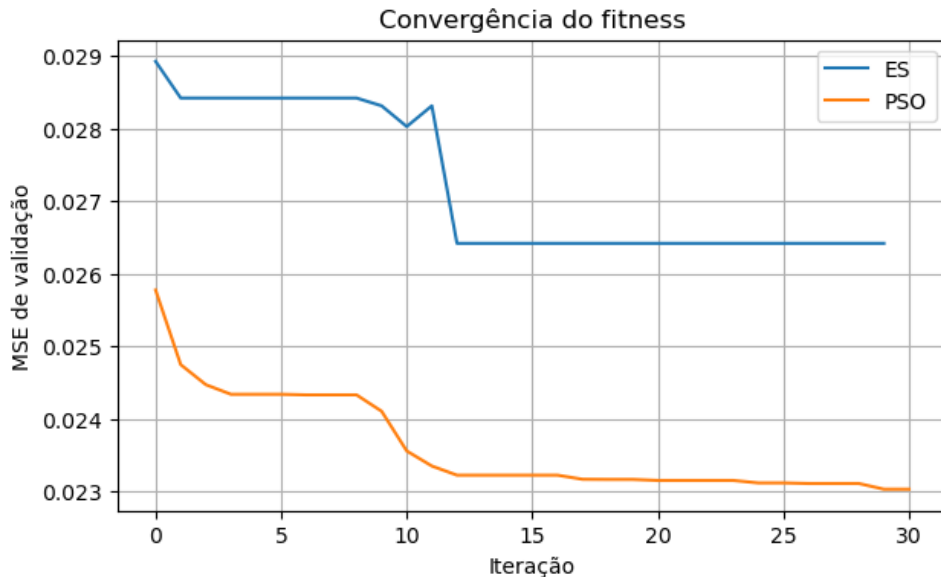


Figura 1: Convergência do fitness (MSE de validação) para os algoritmos ES e PSO. Valores menores indicam melhor desempenho.

A análise dos gráficos revela comportamentos de busca distintos e uma clara diferença de desempenho na otimização da função de aptidão. O PSO exibe uma convergência inicial agressiva, alcançando rapidamente um baixo valor de MSE e continuando a encontrar soluções progressivamente melhores ao longo das 30 iterações, atingindo o menor MSE absoluto de **0.02303**. Em contraste, a ES demonstra uma busca mais contida. Após uma queda inicial, seu progresso estagna em um patamar significativamente mais alto, convergindo para um MSE de **0.02642** por volta da 12ª iteração e não encontrando melhores soluções a partir daí. Fica evidente que, para a tarefa de minimizar o erro de reconstrução, o PSO foi mais eficaz.

3.2 Análise de Desempenho da Classificação

Os melhores indivíduos encontrados por cada algoritmo foram usados para treinar um modelo final, que foi então avaliado no conjunto de teste. A Tabela 2 compara as métricas de desempenho.

Tabela 2: Resultados de desempenho no conjunto de teste para os melhores indivíduos encontrados pela ES e pelo PSO.

Alg.	MSE ↓	Acc ↑	Prec ↑	Rec ↑	F1 ↑	Tempo (s)*	Camadas Ocultas	Taxa Apr.
ES	0.026	0.914	0.915	0.914	0.914	1037	[131, 94, 36]	2.30×10^{-3}
PSO	0.023	0.909	0.908	0.908	0.908	3323	[251, 128, 64]	2.33×10^{-3}

* Tempo médio em 10 execuções com CPU de 14 núcleos.

A análise comparativa revela um *trade-off* notável. O PSO, que se destacou na otimização da função de aptidão, obteve o menor erro de reconstrução (MSE). Isso se deve a ele ter encontrado uma arquitetura de encoder mais profunda e larga, próxima dos limites de busca definidos. No entanto, este foco na fidelidade da reconstrução não se traduziu na melhor performance de classificação.

Surpreendentemente, a ES, apesar de convergir para um MSE maior, produziu um modelo final com desempenho superior em todas as métricas de classificação (acurácia, precisão, recall e F1-Score). A arquitetura mais enxuta encontrada pela ES ([131, 94, 36]) demonstrou uma melhor capacidade de generalização para a tarefa de classificação. Adicionalmente, a ES foi significativamente mais eficiente, concluindo o processo de otimização em um tempo mais de três vezes menor. Este resultado sugere fortemente que a busca agressiva do PSO levou a um sobreajuste (*overfitting*) do autoencoder à tarefa de reconstrução, prejudicando a qualidade das características latentes para a classificação final.

Ao comparar com o artigo de referencia Gilanie et al. (2025) fica evidente que a arquitetura mais enxuta [32, 16, 32] treinada no dataset completo do MNIST, não apenas 15.000 amostras como feito nesse experimento, atingem um desempenho substancialmente superior com 98,85% para GA e 97,77% para Adam (padrão). Dessa forma, é possível concluir que o desempenho do classificador não está diretamente relacionada a uma arquitetura mais complexa do Autoencoder, mas na verdade se trata de um equilíbrio entre o desempenho do Autoencoder e do classificador utilizado.

4 Conclusão

Este estudo demonstrou a eficácia de algoritmos evolutivos na otimização de hiperparâmetros de autoencoders. A metodologia de avaliação em um conjunto de validação interna provou ser crucial para guiar a busca por soluções robustas.

A comparação entre as abordagens revelou um claro *trade-off* entre a otimização da função de aptidão e a generalização do modelo para uma tarefa subsequente:

- **Estratégia Evolutiva (ES):** Apresentou uma convergência para um valor de MSE inferior (para aptidão), mas resultou no modelo com o melhor desempenho de classificação (acurácia de 91.43%). Mostrou-se também computacionalmente mais eficiente, com um tempo de otimização de aproximadamente 1033 segundos. A ar-

arquitetura mais simples encontrada pela ES generalizou melhor para a tarefa de classificação.

- **Otimização por Enxame de Partículas (PSO):** Obteve um erro de reconstrução (MSE de 0.02303) substancialmente menor, explorando arquiteturas de rede mais complexas e se mostrando mais eficaz na otimização da função de fitness. Contudo, essa especialização na reconstrução teve um custo computacional significativamente maior (3323 segundos) e resultou em uma performance de classificação inferior, indicando um sobreajuste à métrica de reconstrução.

Os resultados confirmam o grande potencial da otimização evolutiva para o projeto automatizado de modelos de aprendizado de máquina. A escolha entre ES e PSO depende dos objetivos do problema: se a prioridade é a máxima fidelidade na reconstrução de dados, o PSO é preferível, apesar do custo. Se o objetivo for a eficiência computacional e o melhor desempenho em uma tarefa subsequente como a classificação, a ES se mostrou a abordagem superior neste cenário, mesmo que seu desempenho na função de fitness não tenha sido o melhor.

Referências

GILANIE, Ghulam et al. PARAMETER OPTIMIZATION OF AUTOENCODER FOR IMAGE CLASSIFICATION USING GENETIC ALGORITHM. **Spectrum of Engineering Sciences**, v. 3, n. 4, p. 201–213, 2025.