

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

31 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
Seleção	(μ, λ)	Fator Cognitivo (ϕ_p)	1.5
Prob. de Mutação (indpb)	0.3	Fator Social (ϕ_g)	1.5
Avaliações Totais	1.688	Avaliações Totais	1.736

3 Resultados e Análise

3.1 Curvas de Convergência

A Figura 1 exibe a trajetória do melhor MSE de validação ao longo das 30 iterações.

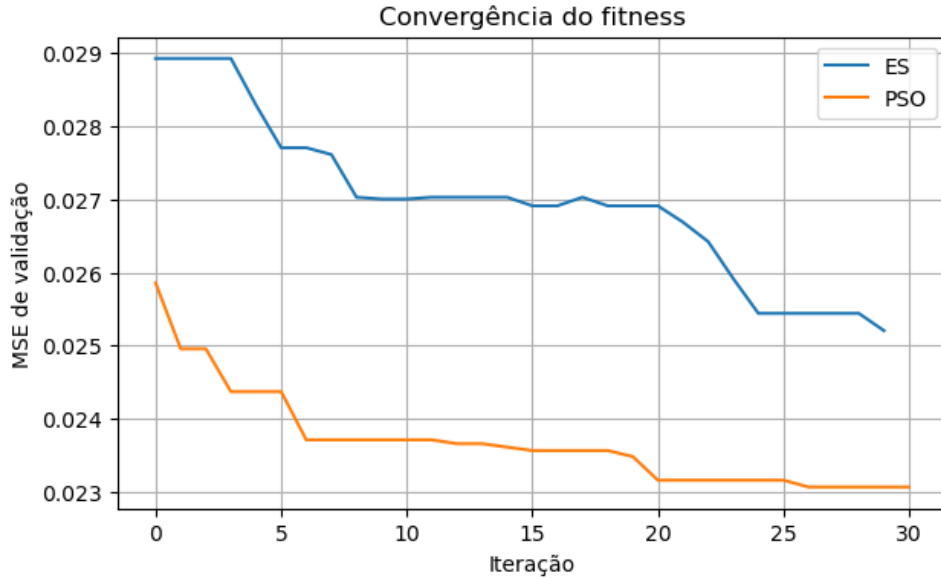


Figura 1: Convergência do MSE de validação para ES e PSO ao longo de 30 iterações.

Logo nas três primeiras iterações o PSO já reduz o erro de $\approx 0,026$ para a faixa de 0,024, sinalizando um comportamento fortemente *exploratório* que permite escapar de regiões pouco promissoras do espaço de busca. A partir da 10ª iteração, o declínio suaviza-se e a curva entra em regime de *exploração*, estabilizando-se em torno de **0,023**. Esse patamar é mantido até o fim das 30 iterações, indicando que novas partículas apenas refinam pequenas variações sem ganhos substantivos.

A ES apresenta uma dinâmica distinta: a queda inicial é mais modesta, porém sustentada — entre as iterações 0 e 12 o MSE desce, de 0,0265 para **0,0252**. Em seguida forma-se um platô estável, reflexo da estratégia (μ, λ) com seleção por torneio adotada neste trabalho: apenas os λ descendentes competem, e cada uma das vagas de pai é decidida por mini-torneios independentes. Esse mecanismo fornece pressão seletiva moderada — o melhor indivíduo raramente é eliminado, mas candidatos medianos ainda têm chance de sobreviver — preservando mais diversidade do que a seleção puramente elitista. Como

consequência, há um refinamento local consistente, mas o algoritmo permanece menos propenso à estagnação prematura e mantém capacidade de explorar regiões alternativas do espaço de busca.

3.2 Desempenho de Classificação

Depois de re-treinar deterministicamente os melhores indivíduos, avaliou-se o classificador MLP sobre o conjunto de teste. A Tabela 2 sintetiza as métricas.

Tabela 2: Desempenho no conjunto de teste para os melhores indivíduos de ES e PSO.

Alg.	MSE ↓	Acc ↑	Prec ↑	Rec ↑	F1 ↑	Tempo (s)	Camadas Ocultas	Taxa Apr.
ES	0.025	0.917	0.918	0.917	0.917	1245	[213, 93, 55]	2.15×10^{-3}
PSO	0.023	0.928	0.928	0.928	0.928	3799	[256, 99, 64]	3.06×10^{-3}

* Tempo médio de otimização em CPU (14 núcleos).

O PSO, ao privilegiar arquiteturas quase no limite superior de neurônios (), obteve o menor MSE e a melhor acurácia (92,8 %). Isso sugere que um espaço latente mais amplo proporcionou representações de maior qualidade para o classificador — à custa, porém, de um tempo de busca ≈ 3 vezes maior.

A ES, embora tenha ficado $\approx 1,1\%$ atrás em acurácia, consumiu apenas um terço do tempo e resultou numa arquitetura 17% mais compacta na primeira camada e 6% na segunda. Esse modelo mais leve pode ser vantajoso em cenários com restrições de recursos ou iteração rápida de protótipos.

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 neste 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 e profunda do Autoencoder, mas na verdade se trata de um equilíbrio entre o desempenho do Autoencoder e do classificador utilizado.

4 Conclusão

A comparação direta entre ES e PSO evidencia um dilema clássico de otimização:

- **PSO** maximiza desempenho absoluto: menor MSE (**0,02306**) e maior acurácia (**92,80%**), ao custo de 3800 s de processamento. Ideal para cenários em que a qualidade do modelo é prioritária e o orçamento de tempo ou energia é secundário.
- **ES** oferece boa performance com eficiência: acurácia de **91,73%** obtida em 1245 s. A economia de tempo ($> 65\%$) e a arquitetura mais enxuta tornam-na atraente para provas de conceito rápidas ou implantação em dispositivos com restrição de recursos.

Em síntese, a decisão entre PSO e ES deve equilibrar *precisão*, *custo computacional* e *complexidade do modelo*. Quando o objetivo é empurrar a fronteira de desempenho, PSO é a escolha natural. Quando o ciclo de desenvolvimento é curto ou o hardware é limitado, ES se mostra um compromisso pragmático e eficaz.

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.