

Documentação do Projeto

Comparação entre Redes Neuro-Fuzzy ANFIS e Redes Neurais MLP

Carlos Gabriel Ribeiro
Afonso Consoli

Junho de 2025

1. Introdução

Este projeto tem como objetivo realizar uma comparação entre dois modelos de aprendizado supervisionado para previsão de consumo energético: uma rede Neuro-Fuzzy do tipo ANFIS (Adaptive Neuro-Fuzzy Inference System) e uma rede neural MLP (Multi-Layer Perceptron). A aplicação foi desenvolvida com **Streamlit**, permitindo ao usuário explorar diferentes configurações de modelos e visualizar os resultados de maneira interativa.

- **Repositório GitHub:** https://github.com/DarkQuantum-bit/MS580_Neuro-Fuzzy
- **Aplicação online:** <https://ms580neuro-fuzzy.streamlit.app/>

Nota: O tempo de treinamento pode ser longo, especialmente para o modelo ANFIS.

2. Bibliotecas Utilizadas

O código utiliza as seguintes bibliotecas:

- `streamlit`: criação da interface web.
- `pandas`, `numpy`: manipulação de dados.
- `matplotlib`: geração de gráficos.
- `scikit-learn`: pré-processamento, divisão dos dados, modelos e métricas.
- `ucimlrepo`: acesso direto ao dataset do UCI.
- `anfis_model`: implementação personalizada da rede ANFIS.

3. Estrutura da Aplicação

A aplicação está organizada em etapas que refletem o pipeline clássico de projetos de aprendizado de máquina:

3.1. Carregamento e Seleção dos Dados

Os dados são obtidos diretamente do UCI Repository, mais especificamente o dataset “Individual household electric power consumption”. O usuário pode escolher um intervalo de datas, as variáveis de entrada e a variável alvo.

3.2. Pré-processamento

- Conversão de colunas `Date` e `Time` para um índice temporal.
- Exclusão de valores nulos.
- Normalização com `MinMaxScaler`.
- Divisão entre treino e teste (`train_test_split`).

3.3. Configuração do Modelo ANFIS

Três perfis de configuração são oferecidos ao usuário:

- **Rápido:** 2 regras, 50 épocas, $LR=0.05$.
- **Balanceado:** 4 regras, 100 épocas, $LR=0.01$.
- **Preciso:** 6 regras, 200 épocas, $LR=0.005$.

3.4. Treinamento dos Modelos

- O modelo ANFIS é treinado com base na implementação própria (`anfis_model.ANFIS`).
- A rede MLP é treinada com `sklearn.neural_network.MLPRegressor`, utilizando 10 neurônios por camada oculta (2 camadas).

3.5. Avaliação e Visualização dos Resultados

As métricas calculadas incluem:

- MAE (Erro Absoluto Médio)
- RMSE (Raiz do Erro Quadrático Médio)
- R^2 (Coeficiente de Determinação)

Além disso, são exibidos:

- Gráficos Real vs Predito para ambos os modelos.
- Curvas de convergência (perda vs épocas).
- Gráficos de resíduos (erros ao longo das amostras).

4. Trecho de Código Destacado

Abaixo, um exemplo de como os modelos são treinados (resumo ilustrativo):

```
# Treinamento do ANFIS
anfis = ANFIS(X_train, y_train, n_rules, len(features))
anfis.train(epochs=epochs, learning_rate=learning_rate)

# Treinamento do MLP
mlp = MLPRegressor(hidden_layer_sizes=(10, 10), max_iter=1,
    ↪ warm_start=True)
for _ in range(epochs):
    mlp.fit(X_train, y_train)
```

5. Considerações Finais

A aplicação é uma ferramenta interativa e educativa para comparar dois paradigmas de modelagem não linear: modelos neuro-fuzzy (ANFIS) e redes neurais convencionais (MLP). A flexibilidade da interface permite realizar experimentos com diferentes variáveis e configurações.

Link para o projeto: https://github.com/DarkQuantum-bit/MS580_Neuro-Fuzzy

Aplicação online: <https://ms580neuro-fuzzy.streamlit.app/>