

Treinamento de Redes Neurais Artificiais Para a Predição de Series Temporais Usando Particle Swarm Optimization

João Gobeti Calenzani

Universidade Federal do Espírito Santo
Vitória, Brazil
joao.calenzani@edu.ufes.br

Victor Nascimento Neves

Universidade Federal do Espírito Santo
Vitória, Brazil
victor.n.neves@edu.ufes.br

Resumo—Este relatório apresenta o desenvolvimento e a análise de modelos de redes neurais treinados para a predição de séries temporais, utilizando algoritmos evolutivos para otimização dos hiperparâmetros (taxa de aprendizado e número de neurônios). Dois conjuntos de dados foram empregados: consumo de energia renovável nos EUA e temperatura da superfície terrestre em relação às mudanças climáticas. O estudo compara o desempenho de uma rede neural otimizada por algoritmos evolutivos com uma rede neural treinada utilizando o algoritmo Particle Swarm Optimization (PSO) no lugar do backpropagation (BP). A eficácia dos modelos foi avaliada pela métrica RMSE (Root Mean Squared Error).

Index Terms—Redes Neurais, PSO, Séries Temporais, Algoritmo Genético

I. INTRODUÇÃO

(Trabalho prático para a disciplina de Computação Natural do programa de Mestrado do Departamento de Informática - UFES. Semestre 2014/01. Prof. Renato Krohling)

Neste estudo experimentamos realizar o treinamento de redes neurais artificiais para a predição de séries temporais, usando o algoritmo Particle Swarm Optimization (PSO) para otimizar os pesos da rede. O objetivo é comparar o desempenho do PSO com o algoritmo de backpropagation, que é o mais comum para treinamento de redes neurais.

O PSO [1] é um algoritmo de otimização global baseado em população, inspirado no comportamento social de bandos de pássaros que se movem em busca de alimento. Cada partícula do PSO representa uma solução candidata no espaço de busca, e a posição da partícula é atualizada iterativamente de acordo com a melhor posição encontrada pela partícula e pela população.

Uma rede neural artificial é um modelo computacional inspirado no sistema nervoso de animais. Consiste em um conjunto de neurônios artificiais organizados em camadas, que são conectados por sinapses. Cada neurônio recebe um conjunto de entradas, realiza uma soma ponderada dessas entradas e aplica uma função de ativação para gerar a saída. O treinamento da rede consiste em ajustar os pesos das sinapses para minimizar o erro entre a saída da rede e a saída desejada.

Comumente, as redes neurais são treinadas usando o algoritmo de backpropagation, que é um algoritmo de otimização

local baseado em gradiente. O algoritmo calcula o gradiente da função de erro em relação aos pesos da rede e atualiza os pesos na direção oposta ao gradiente. O backpropagation é eficiente para redes neurais com muitos pesos, mas pode ficar preso em mínimos locais da função de erro.

Para treinar a rede neural com o PSO, os pesos da rede são codificados como partículas, e a função de aptidão é o erro RMSE da rede para um conjunto de dados de treinamento. O PSO é usado para otimizar os pesos da rede, de forma que a saída da rede seja o mais próxima possível da saída desejada.

Nosso objetivo é comparar o desempenho do PSO com o BP para o treinamento de redes neurais para a predição de séries temporais. Para isso, implementamos o PSO e o BP em Python e treinamos redes neurais para prever as séries temporais do consumo de energia renovável nos USA e da temperatura média global da Terra.

II. METODOLOGIA

A. Treinamento da Rede Neural para Predição de Séries Temporais

A série temporal é uma sequência de valores que representam uma variável ao longo do tempo, denotada por uma variável de tempo t e uma variável $y(t)$, que representa o valor da variável no tempo t . Para treinar a rede neural, utilizamos uma janela deslizante para dividir a série temporal em amostras de treinamento, onde cada amostra consiste em um conjunto de lags da série temporal e o valor da série temporal no próximo instante de tempo.

A camada de entrada da rede possui o mesmo número de entradas que o número de lags da série temporal, e a camada de saída possui uma única saída para prever o próximo valor da série temporal. Experimentamos treinar a rede com 3 e 5 lags da série temporal.

A rede neural contém uma camada oculta, com até 50 neurônios. O número de neurônios na camada oculta é definido através da etapa de otimização de hiperparâmetros. A função de ativação dos neurônios da camada oculta é a ReLU. Essa função de ativação introduz não-linearidade no modelo, permitindo que a rede neural capture padrões complexos na série temporal.

B. Otimização de Hiperparâmetros

Obter os parâmetros ideais para uma rede neural artificial pode ser um processo complexo e dependente da tentativa e erro, o que pode introduzir erros devido ao julgamento do designer da rede. [2] sugere que esse processo pode ser aprimorado utilizando um algoritmo evolutivo para determinar de maneira heurística esses hiperparâmetros.

Neste trabalho, o número de neurônios na camada oculta, assim como a taxa de aprendizado, foram otimizados usando um algoritmo evolutivo, implementado com a biblioteca scikit-optimize (sko). Para implementar a rede neural com backpropagation, utilizamos a biblioteca scikit-learn (sklearn).

Os hiperparâmetros foram codificados em um vetor de quatro dimensões (\mathbf{x}):

$$\mathbf{x} = [x_0, x_1, x_2, x_3]$$

O número de neurônios na camada escondida foi calculado como:

$$\text{hidden_neurons} = 1 + \left\lfloor 50 \times \frac{(x_0 \times 10 + x_1)}{99} \right\rfloor$$

garantindo que hidden_neurons estivesse no intervalo [1, 50].

A taxa de aprendizado inicial foi determinada por:

$$\text{learning_rate_init} = 0.001 + \left(\frac{x_2 \times 10 + x_3}{99} \times (0.01 - 0.001) \right)$$

garantindo que learning_rate_init estivesse no intervalo [0.001, 0.01].

C. Treinamento da Rede com PSO

Após obter os melhores hiperparâmetros para a rede neural, utilizamos o algoritmo Particle Swarm Optimization (PSO) para otimizar os pesos da rede neural. A rede neural treinada com o PSO manteve as mesmas configurações da melhor rede encontrada pela otimização de hiperparâmetros, ou seja, o número de neurônios na camada oculta foi mantido conforme determinado como ideal pelo algoritmo evolutivo.

Implementamos o PSO com topologia local best (lbest). Essa é uma variação do PSO que utiliza uma vizinhança local para atualizar a velocidade e a posição das partículas. A vizinhança é formada por um grupo de partículas que são os melhores vizinhos de cada partícula. Em cada iteração, a partícula verifica se o seu melhor vizinho tem uma posição melhor do que a dela e, se for o caso, atualiza sua posição e velocidade. As fórmulas de atualização da velocidade e da posição das partículas no PSO com lbest são dadas por:

$$v_i(t+1) = \omega v_i(t) + c_1 r_1 (p_i - x_i(t)) + c_2 r_2 (p_{lbest} - x_i(t)) \quad (1)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (2)$$

onde:

- $v_i(t)$ é a velocidade da partícula i no instante de tempo t .

- ω é o fator de inércia que controla a influência da velocidade anterior.
- c_1 e c_2 são os coeficientes de aceleração que determinam a influência do melhor pessoal (p_i) e do melhor local (p_{lbest}).
- r_1 e r_2 são números aleatórios entre 0 e 1.
- $x_i(t)$ é a posição da partícula i no instante de tempo t .
- p_i é a melhor posição já encontrada pela partícula i .
- p_{lbest} é a melhor posição encontrada pela vizinhança local da partícula i .

A implementação do PSO com lbest segue estas fórmulas para atualizar iterativamente as posições e velocidades das partículas, convergindo para uma solução ótima para os pesos da rede neural.

III. EXPERIMENTOS

A. Preparação dos Dados

Os dados utilizados para o treinamento da rede neural são dados tabulares do consumo de energia renovável nos USA ¹ e da temperatura na superfície da terra ²

Na base de dados da temperatura da terra, a variável *LandAverageTemperature* foi utilizada como uma variável endógena, isso é, a variável que se deseja prever. Essa temperatura média é dada mês a mês, e a ideia é prever a temperatura média do mês seguinte. Fig. 1 mostra a série temporal da temperatura média global.

Temperatura Média da Superfície da Terra

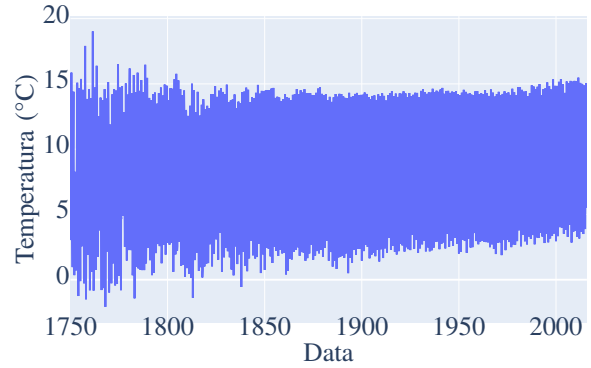


Fig. 1: Temperatura da terra média por mês.

Na base de dados do consumo de energia renovável, a variável *Total Renewable Energy* foi utilizada como uma variável endógena. Ela representa o total de energia renovável (hidroelétrica, geotermal, eólica, solar e de biomassa) consumida nos E.U.A por mês, em trilhões de BTUs. A Fig. 2 mostra a série temporal do consumo de energia.

¹<https://www.kaggle.com/datasets/alistairking/renewable-energy-consumption-in-the-u-s>, acessado em 6 de junho de 2024

²<https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data>, acessado em 6 de junho de 2024

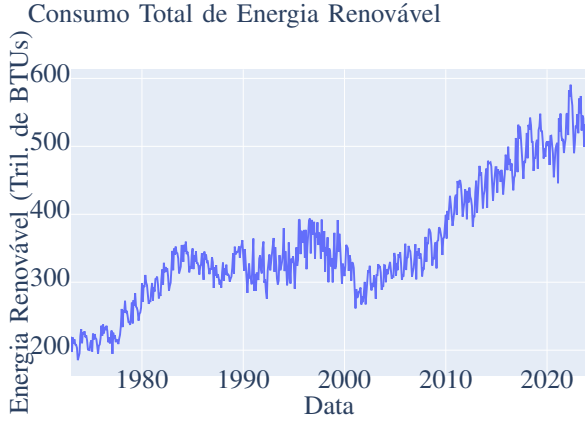


Fig. 2: Consumo de energia renovável por mês.

Para modelar as séries temporais, foram criadas lag features utilizando a variável alvo. Essa técnica permite que o modelo aprenda padrões temporais a partir dos valores anteriores da série. Para um número de defasagens l , foram criadas colunas adicionais representando as temperaturas médias da terra defasadas em l períodos.

Os dados foram divididos em conjuntos de treinamento e teste utilizando a técnica de *TimeSeriesSplit* com 4 divisões. Isso garantiu que o modelo fosse treinado em uma sequência temporal contínua e testado em dados futuros. A divisão resultou em 80% dos dados sendo usados para treinamento e 20% para teste.

Os dados de entrada foram normalizados utilizando a técnica de *StandardScaler*, que ajusta os dados para terem média zero e desvio padrão unitário. A normalização foi aplicada separadamente aos conjuntos de treinamento e teste para evitar data leakage.

B. Otimização dos Hiperparâmetros

A função objetivo do Algoritmo Evolutivo avaliou o desempenho da rede neural para cada conjunto de hiperparâmetros, treinando o modelo e calculando o erro quadrático médio (RMSE) no conjunto de teste:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2}$$

O algoritmo evolutivo foi configurado com uma população de 20 indivíduos e executado por 60 iterações. Foram utilizadas probabilidades de mutação e cruzamento de 0.08 e 0.9, respectivamente. Os limites dos valores dos hiperparâmetros foram definidos como $[0, 9]$ para cada dimensão do vetor.

O algoritmo evolutivo buscou minimizar o RMSE ajustando iterativamente os hiperparâmetros da rede neural. O melhor indivíduo, representando o conjunto ideal de hiperparâmetros, foi selecionado ao final do processo de otimização.

IV. RESULTADOS

A Figura 3 apresenta as curvas de convergência do algoritmo evolutivo durante o treinamento da rede neural com o dataset de temperatura global, utilizando 3 entradas atrasadas ($N=3$) e 5 entradas atrasadas ($N=5$). A Figura 4 exibe a curva de convergência obtida ao treinar com os dados do dataset de energia renovável. Com o resultado dessas rodadas, obtivemos os hiperparâmetros otimizados para treinar as redes neurais. Usamos o número de neurônios na camada escondida para modelar as redes treinadas com backpropagation e PSO. No caso do treino com backpropagation, também obtivemos o parâmetro de taxa de aprendizado.

Fitness das redes de temperatura global

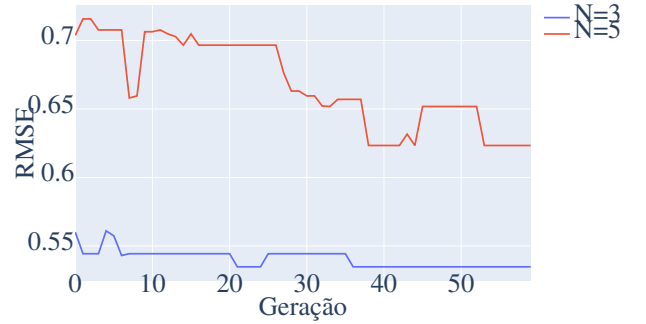


Fig. 3: Curvas de convergência do algoritmo evolutivo para a definição de parâmetros para as redes neurais de previsão da temperatura global. Em vermelho a rede de ordem 3 e em azul a rede de ordem 5.

Fitness das redes de energia renovável

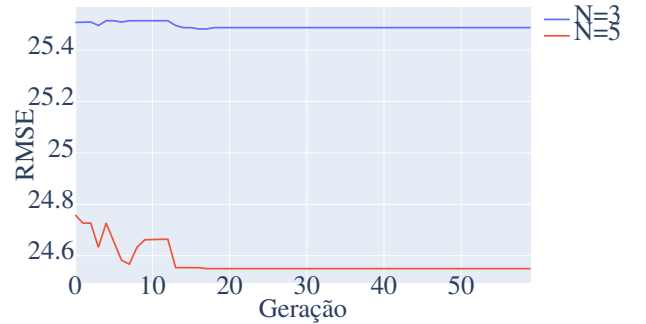
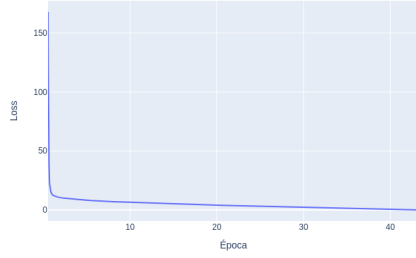


Fig. 4: Curvas de convergência do algoritmo evolutivo para a definição de parâmetros para as redes neurais de previsão do uso de energia renovável.

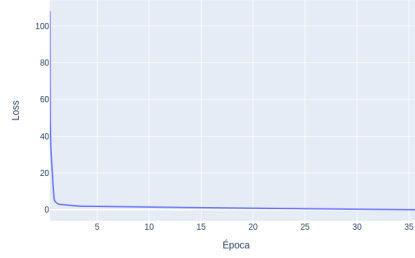
A Tabela I mostra os hiperparâmetros obtidos pelo algoritmo evolutivo para os dois datasets e as duas configurações de entrada (número de neurônios). Esses hiperparâmetros foram utilizados para treinar a rede tanto com backpropagation quanto com PSO.

Mudança Climática (ord. 3)



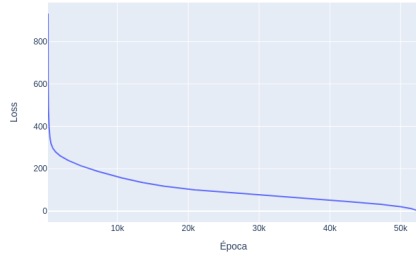
(a) rede de mudança climática de ordem 3

Mudança Climática (ord. 5)



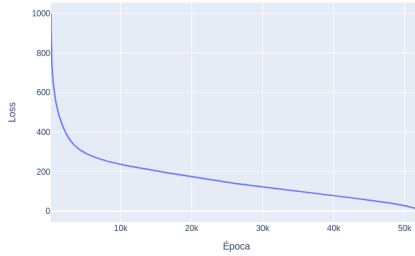
(b) rede de mudança climática de ordem 5

Ener. Renov. (ord. 3)



(c) rede de energia renovável de ordem 3

Mudança Climática (ord. 5)



(d) rede de energia renovável de ordem 5

Fig. 5: Curvas de aprendizado das 4 redes neurais treinadas, usando os melhores hiperparâmetros encontrados.

TABELA I: Hiperparâmetros Otimizados

Dataset: Mudança Climática		
Ordem	Num. Neurônios	Taxa Aprend.
3	29	0.004
5	45	0.008
Dataset: Consumo de Energia		
Ordem	Num. Neurônios	Taxa Aprend.
3	8	0.008
5	5	0.007

A Figura 5 mostra as curvas de aprendizado durante o treinamento das redes neurais com backpropagation. Vemos que as redes de mudança climática convergem rapidamente durante as primeiras iterações enquanto a rede de energia renovável converge mais lentamente. Essa dificuldade de convergência também foi notada durante a fase anterior de fine tuning, onde especialmente para taxas de aprendizado mais baixas, o treinamento não convergia.

A Figura 6 mostra o treinamento das redes neurais usando o PSO. Para cada configuração de rede, 5 rodadas de PSO foram feitas e escolhida a melhor. Nota-se que há uma facilidade maior de convergência das redes de grau 5 em comparação com a de grau 3, sugerindo que com um número maior de inputs, a rede é capaz de aprender os padrões da série temporal mais rapidamente.

Também notamos que as redes convergem consistentemente após cerca de 100 ou menos iterações, inclusive para as

redes de energia renovável com as quais o algoritmo de backpropagation teve mais dificuldade em convergir. Isso pode representar uma capacidade melhor de aprendizado do PSO nesse caso, sendo menos suscetível de cair em mínimos locais, desde que os parâmetros de exploração individual e social sejam bem setados. Nos nossos experimentos, observamos que $\omega=0.7$, $c_1=2.0$ e $c_2=2.0$ deram bons resultados.

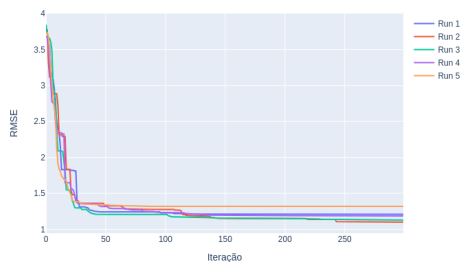
A Tabela II mostra o resultado final das avaliações das redes, com seus respectivos valores de RMSE. Os resultados foram próximos, mas vemos que o backpropagation se saiu um pouco melhor na rede de mudança climática, enquanto o PSO se saiu um pouco melhor na rede de energia renovável.

TABELA II: Resultados

Dataset: Mudança Climática		
Ordem	Método	RMSE
3	Backpropagation	0.5409
3	PSO	0.7130
5	Backpropagation	0.6253
5	PSO	0.7949
Dataset: Energia Renovável		
3	Backpropagation	34.6363
3	PSO	26.1027
5	Backpropagation	24.5495
5	PSO	23.0286

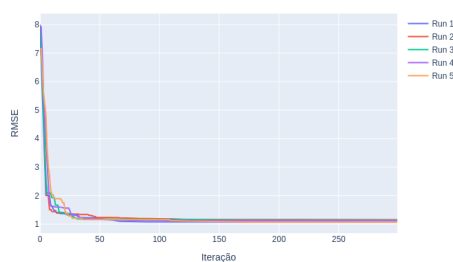
Por fim, as Figuras 7 e 8 mostram os plots dos resultados finais da predição em comparação com os dados reais de teste.

Iterações PSO (Temperatura | N=3)



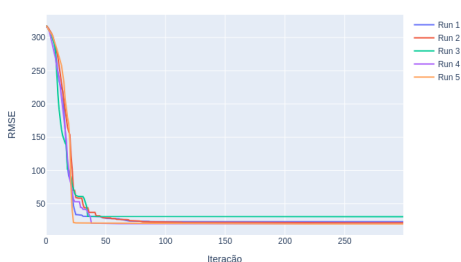
(a) Rede de mudança climática de ordem 3

Iterações PSO (Temperatura | N=5)



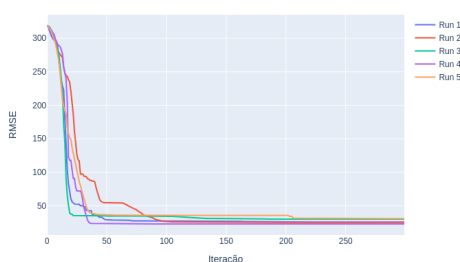
(b) Rede de mudança climática de ordem 5

Iterações PSO (Temperatura | N=3)



(c) Rede de energia renovável de ordem 3

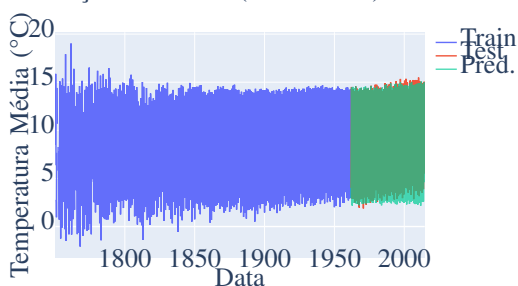
Iterações PSO (Temperatura | N=5)



(d) Rede de energia renovável de ordem 5

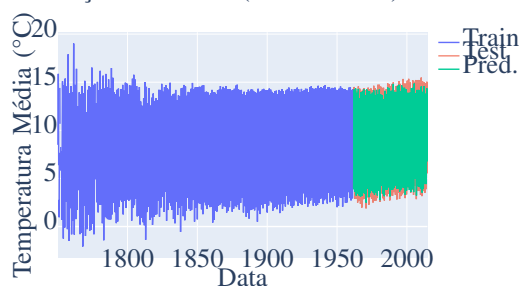
Fig. 6: Curvas de convergência do PSO, em 5 rodadas, das 4 redes neurais treinadas.

Predição Climática (BP — N=3)



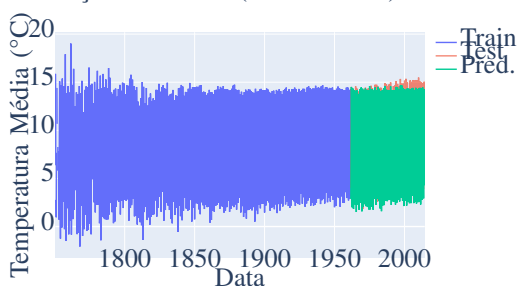
(a) Ordem 3. Treino com BP

Predição Climática (PSO — N=3)



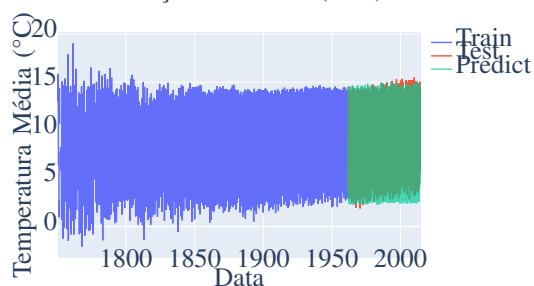
(b) Ordem 3. Treino com PSO

Predição Climática (PSO — N=5)



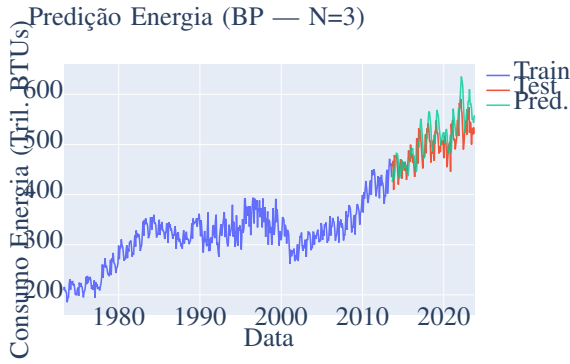
(c) Ordem 5. Treino com BP

BP — Predição Climática (N=5)

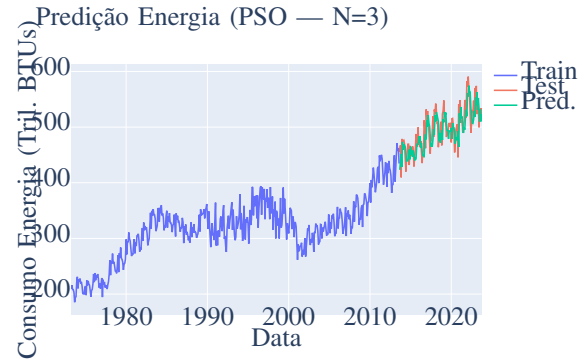


(d) Ordem 5. Treino com PSO

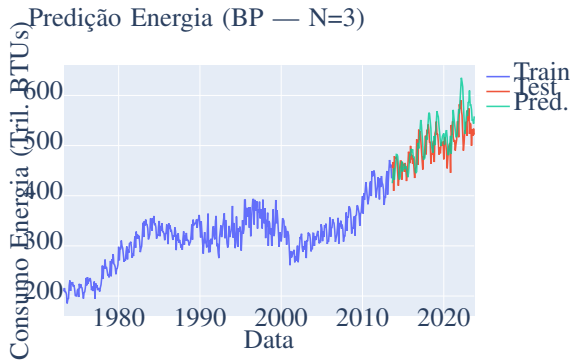
Fig. 7: Predição dos dados de mudança climática. Em azul os dados de treino, em vermelho de teste e em verde, a predição.



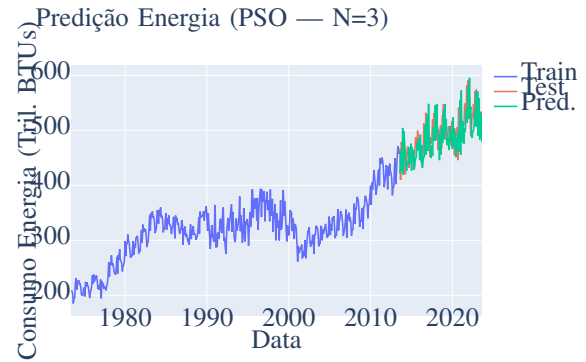
(a) Ordem 3. Treino com BP



(b) Ordem 3. Treino com PSO



(c) Ordem 5. Treino com BP



(d) Ordem 5. Treino com PSO

Fig. 8: Predição dos dados de energia renovável. Em azul os dados de treino, em vermelho de teste e em verde, a predição.

V. CONCLUSÃO

Neste trabalho, comparamos o desempenho de redes neurais treinadas usando backpropagation e Particle Swarm Optimization (PSO) para previsão de séries temporais em dois conjuntos de dados: Mudança Climática e Consumo de Energia Renovável. A análise das curvas de aprendizado revelou que a rede neural treinada com backpropagation convergiu rapidamente para o dataset de Mudança Climática, enquanto a convergência foi mais lenta para o dataset de Consumo de Energia Renovável.

A dificuldade de convergência observada no dataset de Consumo de Energia Renovável foi atribuída à complexidade dos padrões temporais e à escolha das características de atraso (lag features). O algoritmo PSO, por sua vez, demonstrou uma capacidade robusta de aprendizado, convergindo de forma eficiente mesmo para o dataset de Consumo de Energia Renovável, onde o backpropagation encontrou mais desafios.

Os resultados sugerem que o PSO pode ser uma boa alternativa para treinar redes com as características que expetimentamos aqui, relativamente simples e com poucos parâmetros. Ele pode ser especialmente utilizado no caso de convergência difícil do backpropagation, para validar a arquitetura da rede.

Como vimos no caso dos dados de energia renovável, ele foi capaz de aprender os pesos da rede com mais robustez.

REFERÊNCIAS

- [1] James Kennedy and Russell Eberhart. Particle swarm optimization. In *Proceedings of ICNN'95-international conference on neural networks*, volume 4, pages 1942–1948. IEEE, 1995.
- [2] Juan Peralta, German Gutierrez, and Araceli Sanchis. Time series forecasting by evolving artificial neural networks using genetic algorithms and estimation of distribution algorithms. In *The 2010 international joint conference on neural networks (IJCNN)*, pages 1–8. IEEE, 2010.