

Tarefa Computacional 1

Trabalho computacional sobre treinamento de redes neurais usando algoritmos evolutivos para predição de séries temporais

Distribuido em 08 de maio de 2024

Data de entrega: 08 de junho de 2024

Descrição do Problema:

Considere problema de predição de séries temporais como descrito no artigo 1) *Time series forecasting by evolving artificial neural networks using genetic algorithms and differential evolution*. Também pode ser consultado o artigo 2) *Neuro-evolutionary for time series forecasting and its application in hourly energy consumption prediction*. Ambos os artigos estão anexados. Duas séries temporais para realização dos experimentos devem ser usadas (link para os dados no apêndice).

Redes neurais feedforward apresentam uma arquitetura como mostrada na Figura 1. A rede a ser usada deveser conter neurônios na camada de entrada, um número pré-fixado de neurônios na(s) camada(s) intermediária(s) e um neurônio na camada de saída. Usar algoritmos evolutivos para treinar a rede (determinação de pesos de tal forma que a rede neural aprenda o mapeamento entrada-saída). **Deve ser usado o Algoritmo Particle Swarm Optimization (PSO) com topologia lbest, e Differential evolution (DE) com mutação DE/best/1.**

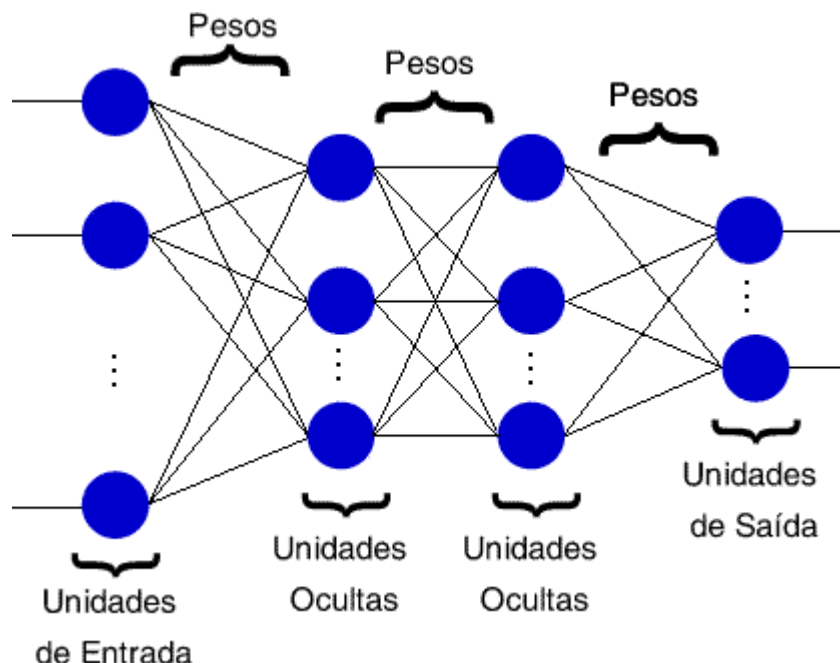


Figura 1: Rede Neural Feedforward

O desempenho da rede é medida pelo critério da raiz do erro quadrático médio (RMSE, root of the mean squared error) ou erro percentual absoluto médio (MAPE, mean absolute

percentage error). Os dados podem ser obtido via download como indicado no apêndice. Separe os dados em padrões de treinamento (80%) e padrões de teste (restante 20%). Considere a ordem do modelo como sendo 3 e depois 5.

O treinamento da rede deverá ser feito usando:

1) scikit-learn com uma camada escondida e no máximo 50 neurônios. Realize a otimização dos hiperparâmetros da rede, e.g. taxa de aprendizado, número de neurônios na camada escondida a fim de obter os melhores resultados.

2) os algoritmos evolutivos PSO e DE. Os pesos devem ser iniciados aleatoriamente. Deverá ser empregada a mesma arquitetura usada no caso 1).

Depois de treinada com os padrões de treinamento, a rede deve fornecer para cada padrão de teste de entrada a correspondente saída do valor previsto. Para analisar os resultados a simulação deve ser executada um certo número de vezes. Por exemplo no mínimo 10 vezes, para que uma estatística possa ser feita em termos de melhor valor obtido (best), a média (mean) do erro de treinamento e teste.

Apresente todos os dados usados nas simulações: Número de neurônios na camada escondida, tamanho da população e número máximo de iterações do algoritmo evolutivo. Plote o valor da fitness em função do número de gerações para análise de convergência do algoritmo evolutivo.

Implementação usando jupyter notebook e python, bibliotecas numpy, scikit-learn etc Faça um relatório sucinto analisando os resultados obtidos em termos de eficácia da predição com as métricas usadas. Envio para email rkrohling@inf.ufes.br com assunto NC-2024-1-Aluno1_Aluno2 (**replica para os exercicios ja enviados**).

Apêndice

Datasets

1) Consumo de energia renovável nos USA

<https://www.kaggle.com/datasets/alistairking/renewable-energy-consumption-in-the-u-s>

2) Mudança climática - temperatura na superfície da terra

<https://www.kaggle.com/datasets/berkeleyearth/climate-change-earth-surface-temperature-data>