Daily streamflow forecasting in Sobradinho Reservoir using machine learning models coupled with wavelet transform and bootstrapping

Samuel Vitor Saraiva, Frede de Oliveira Carvalho, Celso Augusto Guimarães Santos, Lucas Costa Barreto, Paula Karenina de Macedo Machado Freire

Apresentação: Rodrigo Cunha

São Paulo, 14 de dezembro de 2021

Tópicos

1. Introdução
2. Objetivos
3. Transformada discreta de ondaletas (DWT)
4. Redes neurais artificiais (ANN)
5. Experimentos
6. Dados
7. Resultados
8. Conclusões

1. Introdução

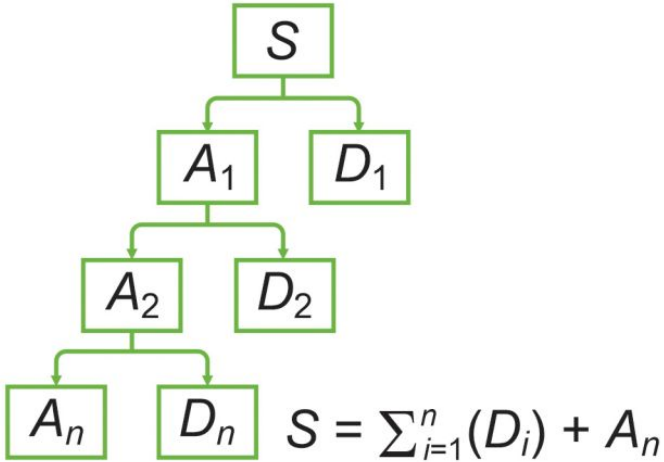
- ***Daily streamflow forecasting in Sobradinho Reservoir using machine learning models coupled with wavelet transform and bootstrapping***, Applied Soft Computing Journal (2021)
- Pesquisadores brasileiros (UNICAMP, UFAL, UFPB)
- Problema de hidrologia
  - Previsão da vazão de água no reservatório da usina de Sobradinho, BA
- Dados de vazão obtidos da ONS (Operador Nacional do Sistema, http://www.ons.org.br)
  - Medidas diárias de vazão em m^3/s, compreendendo o período de 1931 a 2015.
    - **Treino**: 1931 a 2004
    - **Teste**: 2005 a 2015
- Os autores realizam testes com modelos MLP e SVM. Os resultados com e sem a decomposição com DWT são comparados, indicando que a decomposição melhora o desempenho dos modelos significativamente.

2. Objetivos

- Utilizar a transformada discreta de ondaletas (DWT) para decompor a série de vazão
- As séries decompostas são utilizadas como input de modelos de *machine learning* (ML)
    - *Artificial Neural Network* (ANN) e *Support Vector Machine* (SVM)
- O objetivo é expandir o espaço de treinamento dos modelos de modo a melhorar o desempenho na previsão da vazão

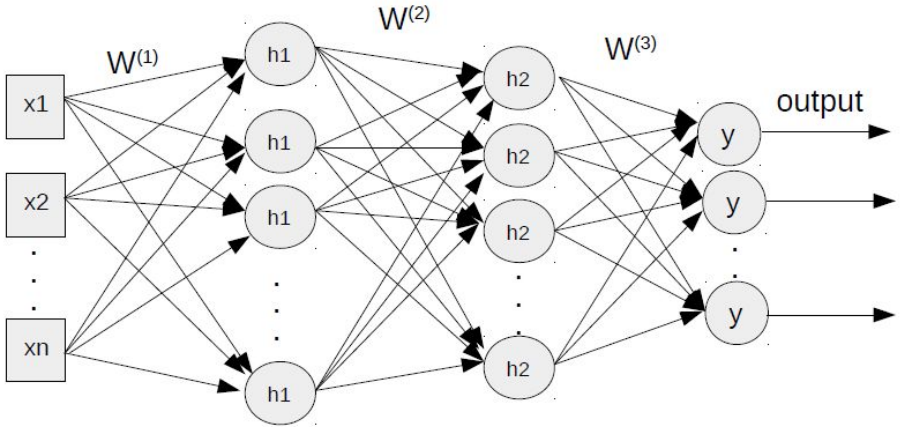
3. Transformada discreta de ondaletas (DWT)

A decomposição da série original permite que diferentes dinâmicas presentes no sinal original sejam analisadas separadamente pelos modelos de previsão.



4. Redes neurais artificiais (ANN)

As redes neurais artificiais são modelos de aprendizado de máquina supervisionado inspirados no funcionamento do neurônio biológico. Em um modelo ANN com arquitetura MLP (*Multi-layer Perceptron*), cada nó corresponde a um *neurônio*, que recebe o somatório das saídas dos neurônios da camada anterior ponderados por um vetor de pesos e o submete à uma função de ativação, gerando a sua saída.



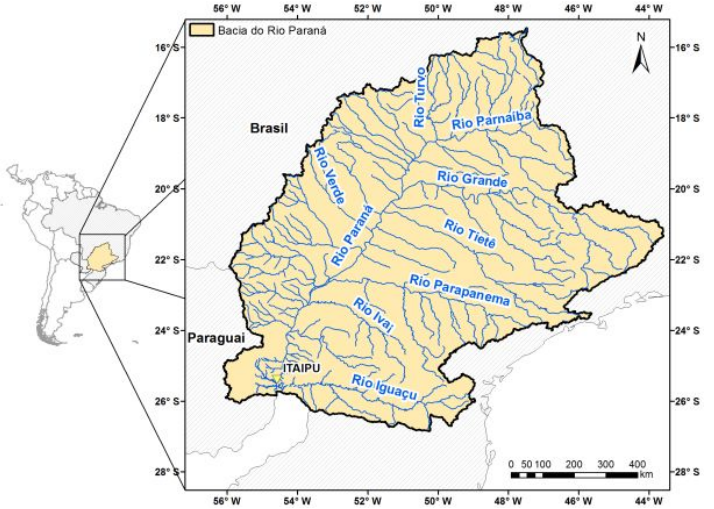
4. Redes neurais artificiais (ANN)

- Atingem bons resultados em uma ampla variedade de problemas
- Modelos flexíveis, podendo ser aplicados para problemas de classificação e regressão
- Difíceis de treinar: alta demanda por dados, tempo e capacidade computacional
- Necessidade de ajustar hiperparâmetros manualmente
- A arquitetura MLP não é a mais adequadas para problemas envolvendo séries temporais
    - Arquiteturas como as redes neurais recorrentes (RNN) obtém melhores resultados

5. Dados

Os dados utilizados nos experimentos são fornecidos pela ONS (Operador Nacional do Sistema), e correspondem a medidas diárias da **energia natural afluente** para o período de 2000 a 2021 na bacia do Rio Paraná.

- A energia natural afluente (ENA) corresponde à quantidade de água disponível que pode ser utilizada para produção de energia. Consiste basicamente no valor da vazão total da bacia multiplicada por uma constante que incorpora a capacidade das usinas.

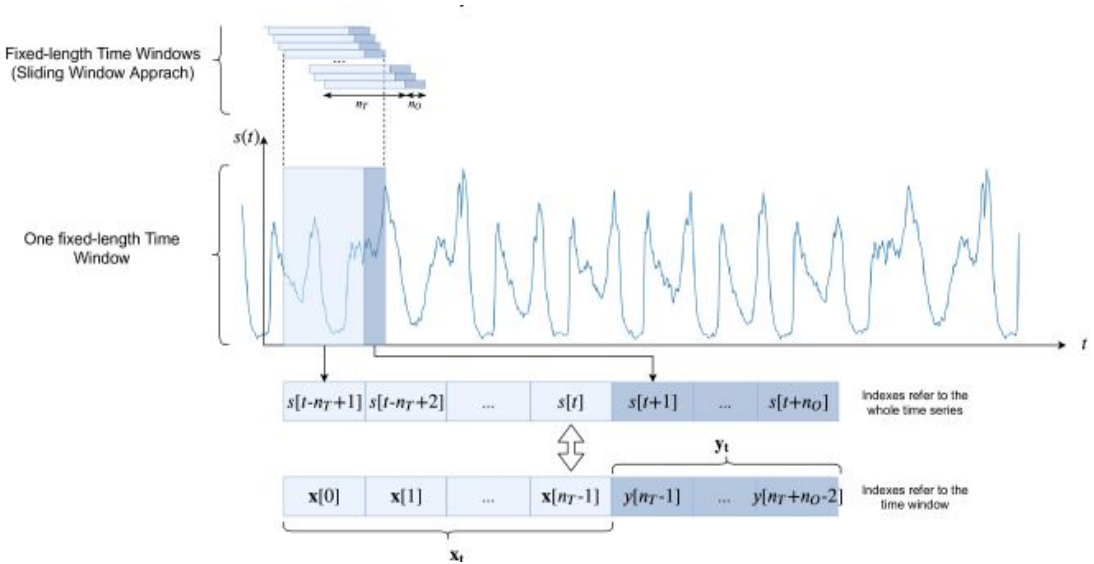


5. Dados

O ajuste dos parâmetros de uma rede MLP depende de **treinamento supervisionado**. Ou seja, são necessários pares **input-output** que devem ser utilizados como exemplos durante o treinamento da rede.

Os dados brutos correspondem a uma única série, que vai de 2000 a 2021. Para construir os pares **input-output** é necessário **janelar** a série original.

O janelamento da série foi realizado considerando um tamanho de 145 pontos (128 + 15).



5. Dados

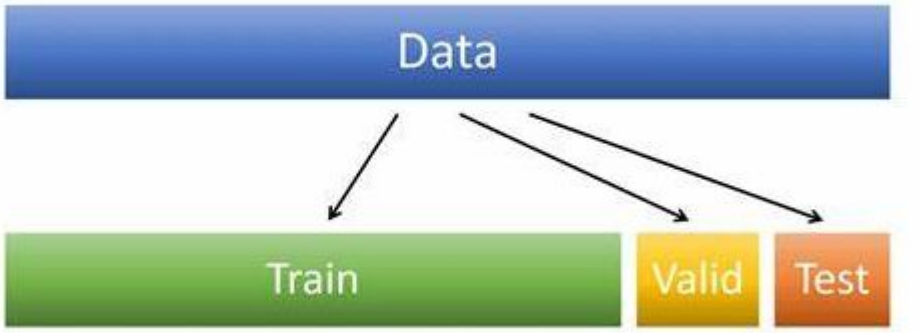
Com o janelamento da série original, se obtém os pares **input-output** necessários para o treinamento da rede. O treinamento da rede exige ao menos três datasets com finalidades distintas:

- **Dataset de treino**: o dataset de treino é utilizado para calcular os gradientes necessários para estimar os pesos do modelo. Durante o treinamento, o modelo recebe um **input** e devolve uma **previsão**. A partir da diferença entre o valor previsto pelo modelo e o valor correto, calculam-se os gradientes necessários para estimar os pesos do modelo. Ou seja, o treinamento age no sentido de diminuir o erro do modelo.
- **Dataset de validação**: o dataset de validação é utilizado durante a etapa de treinamento, porém o modelo não os utiliza ativamente no treinamento. A cada época de treinamento, o dataset de validação é utilizado para testar o modelo e avaliar a convergência do modelo (casos de **overfitting**, por exemplo). Em outras palavras, o dataset de validação não é utilizado para calcular gradientes, apenas para avaliação.
- **Dataset de treinamento**: após realizado o treinamento do modelo, seu desempenho é avaliado diante de um dataset com exemplos desconhecidos, de modo a avaliar a generalização do modelo.

5. Dados

Em geral, utiliza-se de 70 a 80% do dataset original para o **dataset de treino**, 10% para o **dataset de validação** e 10% para o **dataset de test**.

Nesse caso, o **split** dos dados respeitou a ordem cronológica da série temporal, de modo que os dados mais recentes foram utilizados para validação e teste, enquanto os dados mais antigos foram utilizados para treino. O objetivo é reproduzir uma situação real, em que os dados de **teste** não são conhecidos.



5. Dados

Após o **janelamento** e **split**, é necessário fazer o **scaling** dos dados. Modelos ANN são dependentes da escala dos inputs, e atingem desempenhos melhores quando os dados são escalados para valores próximos de 1.

Os dados foram escalados utilizado o **z-score.** Para calcular a média e o desvio padrão necessários para realizar o cálculo, apenas do dataset de treino foi considerado.

$$Z = \frac{x - \mu}{\sigma}$$

Z = standard score

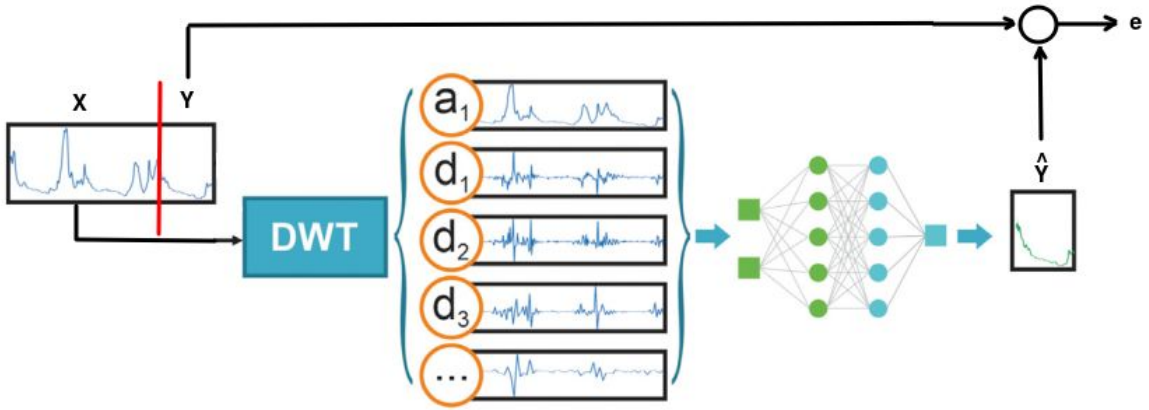
x = observed value

μ = mean of the sample

σ = standard deviation of the sample

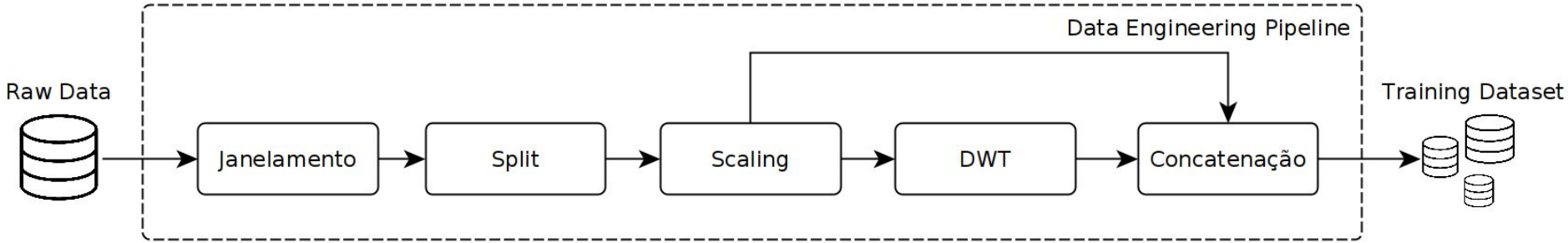
5. Dados

Finalmente, após as etapas de **janelamento**, **split**, e **scaling**, podemos calcular a DWT para os dados de input (primeiros 128 pontos das janelas). Para compor o vetor de entrada, podemos concatenar a série temporal com as componentes DWT ou não, ou mesmo utilizar apenas a série temporal como vetor de entrada (caso canônico).



5. Dados

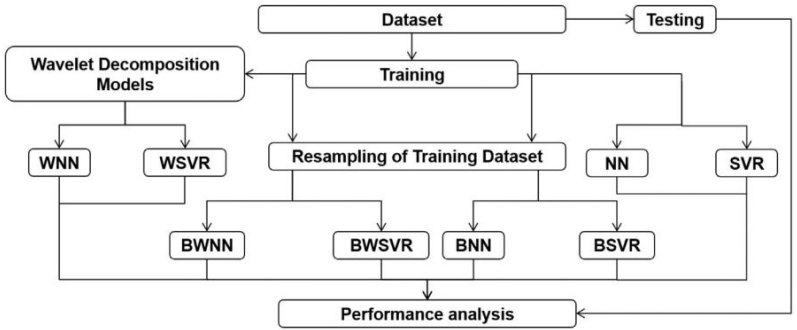
O **pipeline de data engineering** está representado pelo diagrama abaixo:



6. Experimentos

Os experimentos conduzidos pelo autor do artigo avaliam o desempenho dos modelos MLP (um tipo de ANN) e SVM em diferentes configurações:

- Com e sem a decomposição utilizando ondaletas
- Com e sem a técnica de *bootstraping*



6. Experimentos

Porém, neste seminário apenas o modelo MLP foi considerado, levando em consideração as seguintes variações:

- MLP sem DWT (apenas a série temporal)
- MLP com DWT mais a série temporal
- MLP com DWT (apenas as componentes da DWT)

Todos os códigos utilizados no projeto estão disponíveis no Github:

➔ https://github.com/rodra-go/enaforecast

7. Conclusões

- A decomposição da série de entrada utilizando a transformada DWT de fato favorece o desempenho da rede, porém de forma marginal
- O volume reduzido de dados não permite que a rede atinja um desempenho satisfatório
  - Apenas 8011 pontos no dataset original
    - No meu projeto de pesquisa, os datasets têm (no mínimo) 50 milhões de linhas
      - Cerca de 6300 vezes maior
- Para que um modelo MLP atinja bons resultados, é necessário ajustar os hiperparâmetros
  - O ajuste é realizado a partir de tentativa e erro
    - Técnicas mais modernas utilizam modelos de otimização para encontrar hiperparâmetros ótimos
- Prever séries temporais é um problema extremamente complexo. Na prática, nenhum modelo é capaz de resolver esse problema facilmente.
  - Diversos ajustes serão necessários, independente do modelo escolhido
- Não fica claro se o autor treinou o modelo para prever um único ponto ou diversos pontos (como no caso dos meus experimentos)

Fim.

Obrigado pela atenção!