



**Centro de
Informática**
UFPE



UNIVERSIDADE
FEDERAL
DE PERNAMBUCO

Modelos de Redes Neurais como Proxy para Simulador de Reservatórios

Johnattan Douglas
Juan Tueros
Victor Borges

03 de Julho de 2024

1. Motivação
2. Introdução
3. Proposta
4. Metodologia
5. Experimentos
6. Conclusões

Motivação - Problema de injeção de água

A injeção de água é o método mais comum para a recuperação de óleo (1890).

Principais características:

- Incertezas nas propriedades;
- Múltiplos reservatórios;
- Milhares de poços;
- Não linearidade.

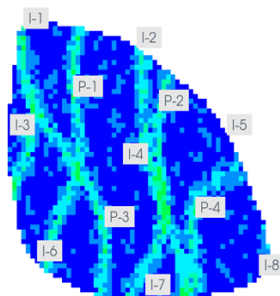


Figura 1: Egg model

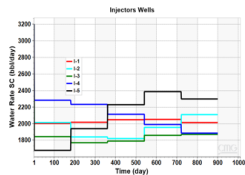
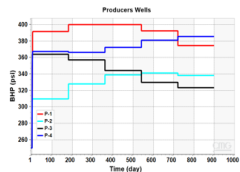
Portanto temos:

Alto custo computacional em um único processo de simulação.

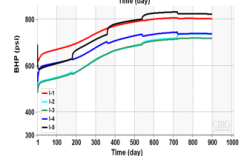
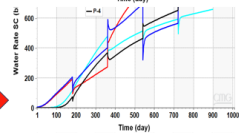
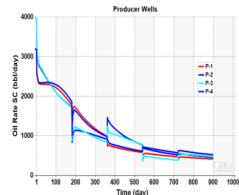
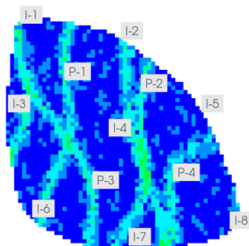
Atuação do Simulador



IMEX 2023.10



Inputs



Outputs

Na engenharia de reservatórios, os dados do histórico de produção são abundantes.

Proposta

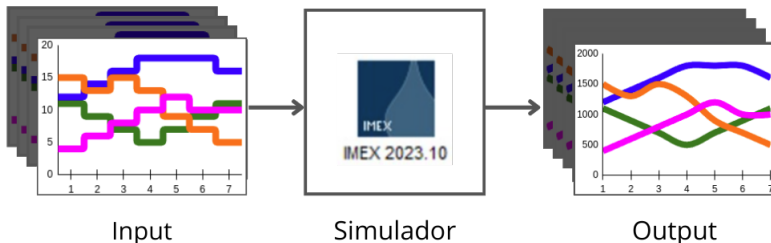
Na indústria do petróleo, os dados do histórico de produção e injeção, tem o comportamento de uma série temporal devido à mudança de controles das entradas.

1. Utilizar os dados do histórico de produção para que as redes neurais aprenda a **DINÂMICA** do simulador;
2. Identificar as melhores técnicas para a previsão do histórico de produção de poços produtores e injetores, **comparando as técnicas LSTM e GRU e incorporando o mecanismo de atenção** para reduzir a variabilidade dos resultados.
3. Determinar os melhores hiperparâmetros das arquiteturas utilizando a estratégia de busca em grade (grid search);
4. Aplicar a metodologia a problemas existentes na literatura.

Geração do Banco de dados

- Os controles dos poços produtores e injetores foram gerados utilizando a função gaussiana para gerar controles correlacionados. Os controles são mudados a cada 180 dias e as respostas recuperadas a cada 30 dias em 1800 dias de simulação.
- As simulações dos controles foram executados no simulador de reservatórios IMEX (Implicito - Explícito).
- O dataset é formado pelos controles dos poços produtores e injetores, e pelos respostas de produção de óleo, produção de água e pressão dos poços injetores.
- O número de amostras total de amostras é 1463, as quais serão utilizadas para treinamento (70 %), validação (20 %) e teste (10 %).

Atuação do simulador



- **Inputs:** Controles de pressão nos poços produtores e vazão nos poços injetores
- **Outputs:** Produção de óleo, produção de água e pressão no poço injetor.

Metodologia - Long Short Term Memory - LSTM

LSTM

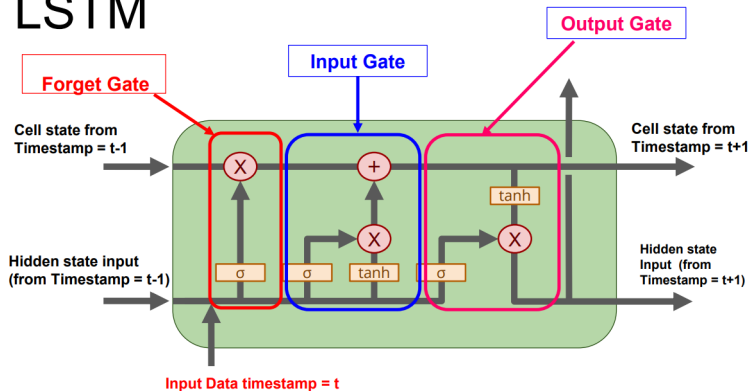


Figura 2: Long Short Term Memory

LSTM (Forget gate)

- **Objetivo:** Decidir qual informação deve ser armazenada ou “esquecida”.
- **Entrada:**
 - Hidden state do passo de tempo anterior.
 - Input do passo de tempo atual.
- **Saída:** Valores próximos a 0 indicam informações que devem ser “esquecidas”, enquanto valores próximos a 1 indicam informações que devem ser armazenadas.

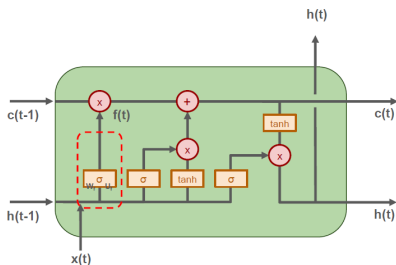


Figura 3: Forget gate

LSTM (Input gate)

- **Objetivo:** Atualizar o cell state.
- **Entrada:** hidden state do passo de tempo anterior e input do passo de tempo atual.
- **Saída:** A saída é determinada pela combinação de dois aspectos:
 - Um componente que avalia a relevância da informação anterior.
 - Outro componente que normaliza os valores para melhorar o ajuste da rede neural.

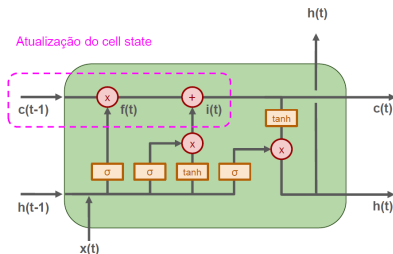
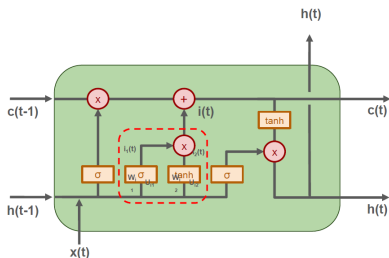


Figura 4: Input Gate

LSTM (Output gate)

- **Objetivo:** Calcular o valor do próximo hidden state (output).
- **Entrada:** hidden state do passo de tempo anterior e input do passo de tempo atual.
- **Saída:** Para calcular o próximo estado oculto:
 - Primeiro, calculamos $o(t)$, que determina a relevância do estado oculto anterior e do input atual.
 - Em seguida, utilizamos $o(t)$ para modular o estado da célula através da função tangente hiperbólica, resultando no próximo estado oculto.

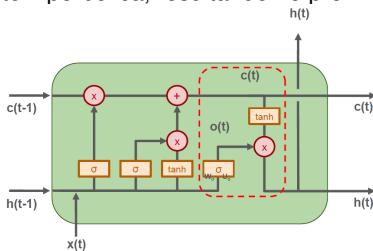


Figura 5: Output gate

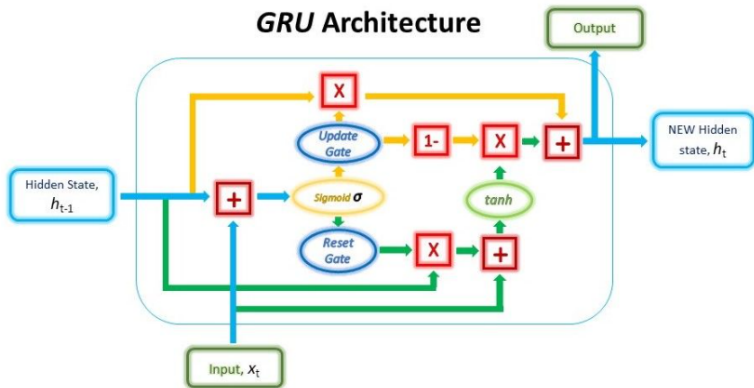


Figura 7: Arquitetura GRU

LSTM vs GRU

- GRUs utilizam apenas um estado oculto;
- LSTMs possui dois estados distintos (o estado da célula e o estado oculto) para carregar memórias de longo e curto prazo;
- O estado oculto da GRU é transferido entre as etapas de tempo e consegue reter dependências tanto de longo quanto de curto prazo.

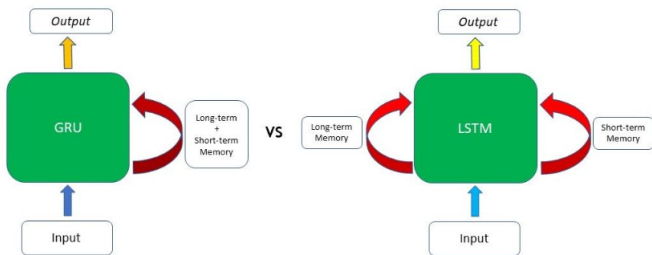


Figura 8: LSTM vs GRU

- Nas RNNs convencionais, cada input corresponde a um output para um mesmo instante de tempo;
- Entretanto, muitas vezes queremos prever um output sequence dado um input sequence de tamanho diferente(ex. tradução, chatbots, dispositivos ativados por voz etc.);
- Tais modelos são conhecidos por sequence-to-sequence mapping;
- O modelo Encoder-Decoder para RNNs foi então desenvolvido a fim de lidar com modelos desse tipo.

- O modelo Encoder-Decoder pega uma sequência como input e gera a próxima sequência mais provável como output.

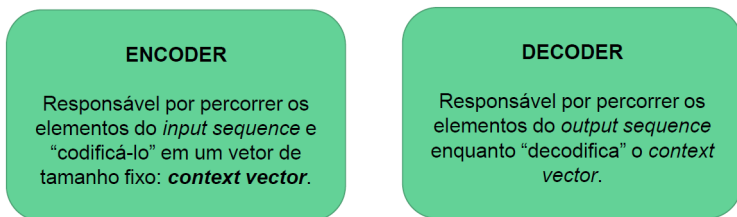


Figura 9: Encoder-Decoder

Attention

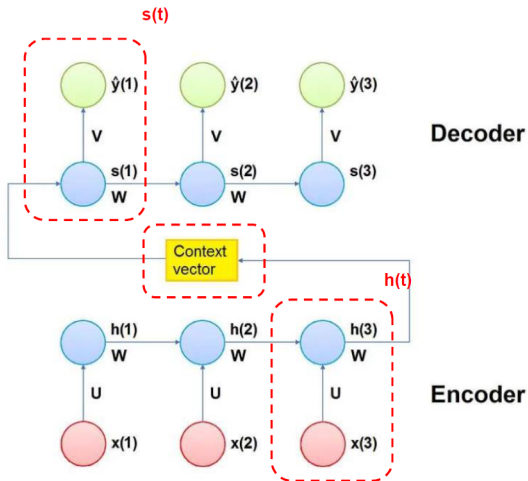


Figura 10: Arquitetura Encoder-Decoder

Attention

- É uma evolução do modelo Encoder-Decoder, desenvolvido a fim de melhorar a sua performance diante de longas sequências de entrada;
- A ideia principal é permitir o decoder acessar seletivamente informações do encoder durante a “decodificação” (processo de geração de outputs $y^{(t)}$);
- Isso pode ser feito construindo um context vector diferente para cada instante de tempo do decoder, obtido em função do hidden state anterior $s(t-1)$ e de TODOS os hidden states do encoder $h(t)$;
- Dessa forma, o mecanismo de atenção atribui importância diferente para diferentes elementos do input sequence, e dará mais “atenção” aos inputs mais importantes.

Attention

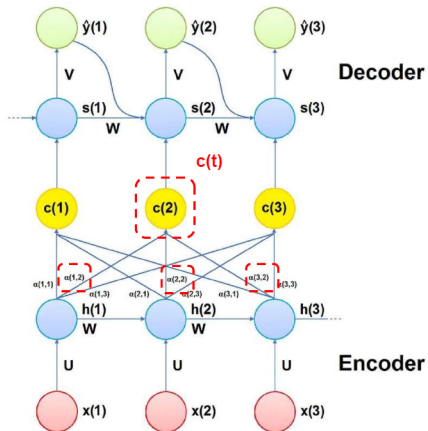
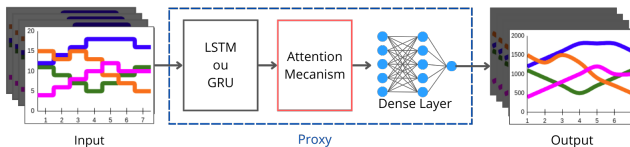
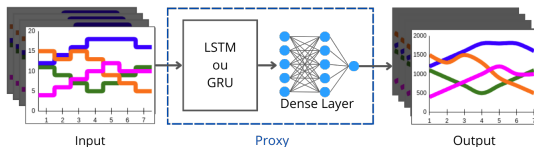
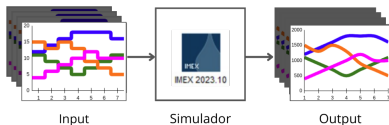


Figura 11: Arquitetura Encoder-Decoder Attention

Propostas de Proxy



Experimentos

- Divisão dos dados: 70/20/10
- Métrica de Avaliação: MAE (treinamento) e RMSE (teste)
- Otimização de Hiperparâmetros
 - Número de unidades por camada: 100 a 400, de 50 em 50
 - Tamanho de batch = 4, 8, 16, 32, 64, 128 e 256
 - Taxa de aprendizagem: 10^{-5} , 10^{-4} , 10^{-3}
 - Função de ativação: Linear, ReLu e Tanh
- Ambiente de execução: Apuana

Hiperparâmetros

Hiperparâmetro	LSTM	GRU
Número de Neurônios	400	350
Tamanho do Batch	4	4
Taxa de Aprendizagem	10^{-3}	10^{-3}
Função de Ativação	ReLU	Linear

Tabla 1: Hiperparâmetros após grid search.

Função de Ativação

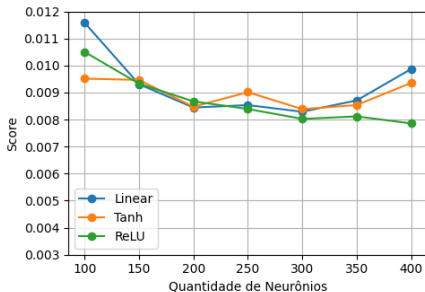


Figura 12: LSTM

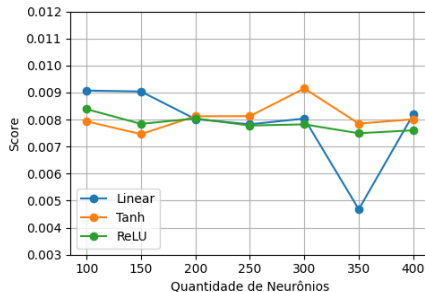


Figura 13: GRU

Comparação entre modelos

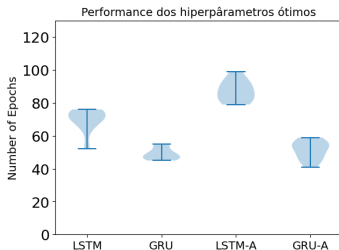


Figura 14: Número de Épocas

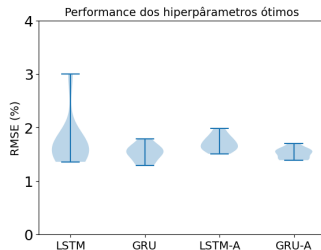
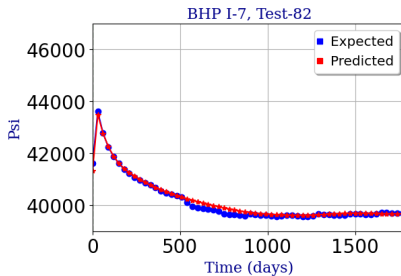
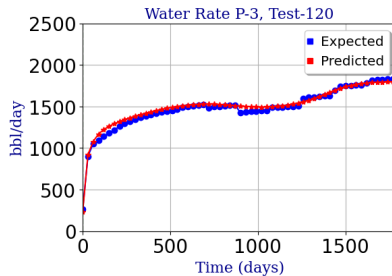
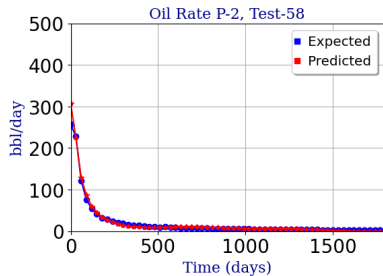


Figura 15: RMSE

	Parâmetros treináveis	Nº de Épocas	RMSE %
LSTM	667216	70	1.72
LSTM-A	667216	87	1.71
GRU	387816	62	1.53
GRU-A	387816	74	1.51

Tabla 2: Comparação das estratégias propostas

Previsão



Conclusões

1. Proposta de um modelo proxy baseado em redes neurais para **prever as respostas de um simulador de reservatórios**, vazão de óleo e água nos produtores e pressão dos poços injetores.
2. Foi possível **aprender a dinâmica do simulador** de reservatório!!!
3. ReLU é mais eficaz quando utilizada com uma maior quantidade de unidades;
4. O mecanismo de atenção reduz a variabilidade dos resultados, embora aumente o custo computacional.
5. GRU-A como modelo proxy apresentou o melhor desempenho ao prever as respostas dos poços produtores e injetores.



**Centro de
Informática**
UFPE



UNIVERSIDADE
FEDERAL
DE PERNAMBUCO

Modelos de Redes Neurais como Proxy para Simulador de Reservatórios

Johnattan Douglas
Juan Tueros
Victor Borges

03 de Julho de 2024