



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA CIVIL E AMBIENTAL
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA CIVIL

LETÍCIA AGRA MENDES RAMALHO

**TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA CARACTERIZAÇÃO DAS
CONECTIVIDADES ENTRE POÇOS EM RESERVATÓRIOS DE PETRÓLEO**

Recife
2023

LETÍCIA AGRA MENDES RAMALHO

**TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA CARACTERIZAÇÃO DAS
CONECTIVIDADES ENTRE POÇOS EM RESERVATÓRIOS DE PETRÓLEO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal de Pernambuco como requisitos para obtenção do título de Mestre em Engenharia Civil.

Área de concentração: Simulação e Gerenciamento de Reservatórios de Petróleo.

Orientador: Prof. Dr. Bernardo Horowitz.

Coorientador: Dr. Juan Alberto Rojas Tueros.

Recife

2023

Catálogo na fonte
Bibliotecária Margareth Malta, CRB-4 / 1198

R166t	Ramalho, Letícia Agra Mendes. Técnicas de inteligência artificial para caracterização das conectividades entre poços em reservatórios de petróleo / Letícia Agra Mendes Ramalho – 2023. 102 f.: il., figs., qds., tabs. Orientador: Prof. Dr. Bernardo Horowitz. Coorientador: Dr. Juan Alberto Rojas Tueros. Dissertação (Mestrado) – Universidade Federal de Pernambuco. CTG. Programa de Pós-Graduação em Engenharia Civil, 2023. Inclui Referências e Anexos. 1. Engenharia Civil. 2. aprendizado de máquina. 3. Convolutional neural networks. 4. Embaralhamento. 5. Conectividades de poços. 6. Simulação de reservatório. I. Horowitz, Bernardo (Orientador). II. Rojas Tueros, Juan Alberto (Coorientador). III. Título. UFPE 624 CDD (22. ed.) BCTG/2024-71
-------	---

LETÍCIA AGRA MENDES RAMALHO

**TÉCNICAS DE INTELIGÊNCIA ARTIFICIAL PARA CARACTERIZAÇÃO DAS
CONECTIVIDADES ENTRE POÇOS EM RESERVATÓRIOS DE PETRÓLEO**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Civil da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para a obtenção do título de Mestre em Engenharia Civil. Área de concentração: Simulação e Gerenciamento de Reservatórios de Petróleo.

Aprovada em: 06/11/2023.

BANCA EXAMINADORA

Prof. Dr. Bernardo Horowitz (Orientador)
Universidade Federal de Pernambuco - UFPE

Profa. Dra. Silvana Bastos Afonso da Silva (Examinador Interno)
Universidade Federal de Pernambuco - UFPE

Profa. Dra. Licia Mouta da Costa (Examinador Externo)
Universidade Federal da Paraíba – UFPB

Profa. Dra. Leila Brunet de Sá Beserra (Examinador Interno)
Universidade Federal de Pernambuco - UFPE

Dedico este trabalho ao meu pai, Edson Romero, que sempre me inspirou e me ajudou a seguir a carreira acadêmica, mesmo em seus últimos instantes de vida; e minha mãe, Lêda Ramalho, que encontra-se no céu junto ao meu pai.

AGRADECIMENTOS

Agradeço primeiramente a Deus, O qual foi minha fortaleza durante toda essa jornada, que muitas vezes me resgatou em momentos de fraqueza e me fez forte, me fez amada e avançar cada vez mais nessa caminhada. À minha intercessora, a minha mãe, que só nós órfãos sabemos o valor e o amor que ela sente por nós, Nossa Senhora, Maria mãe de Jesus, que sempre foi socorro nos momentos de angustia e desespero.

A minha família, a qual amo e sou grata por tudo, quem me abriu as portas e me fez chegar onde cheguei. Ao meu pai, que mesmo em momentos de desespero e medo, se preocupava com minha vida acadêmica, sempre me lembrando da importância do meu mestrado e do que venho galgando ao longo dos anos, principalmente ao meu pai, por ter virado meu anjo durante essa trajetória. A minha mãe, querida, que sempre me mostrou a importância de estudar, que sempre dedicou minutos dos seus dias pra sentarmos juntas e aprender um pouco mais.

Ao meu psiquiatra, Doutor Kaikê Thiê, que foi luz nesses quase dois anos de atendimentos, que me resgatou de um poço no qual achava que não tinha fundo e me fez enxergar muito mais força e dedicação para que eu pudesse continuar.

A minha irmã, Luana, a quem protejo e quero bem todos os dias e tenho a certeza também por meio de suas atitudes da reciprocidade da nossa irmandade. Quem sempre esteve comigo nas viagens a Recife para decisões importantes da dissertação e tornou a caminhada ainda mais leve, ao meu cunhado, que também sempre esteve presente nos momentos de maiores tensões. Aos meus irmãos, Laura e Romerinho, que proporcionaram momentos de alegria e aconchego em meio aos momentos onde eu me encontrava exageradamente focada.

Ao meu noivo, Maércio, que também foi peça chave nessa caminhada, sempre me dando força e dizendo o quanto eu era capaz, me incentivando e me fazendo ter cada vez mais interesse pela minha escrita e o progresso da minha caminhada.

Ao professor Bernardo, que desde o princípio abraçou essa proposta bem difícil de me ter como orientanda, que sempre foi prestativo e me mostrou o quanto um professor pode ser ótimo em sua profissão e didática.

Ao meu coorientador Juan, que sempre me estendeu a mão e me ajudou em momentos de angustia, que andou comigo nas fases do luto e o que vem depois e mesmo assim conseguimos juntos concluir mais uma etapa da minha vida acadêmica.

Agradeço à FACEPE, pelo incentivo financeiro em mim investido para a realização dessa pesquisa.

Ademais, agradeço a todos que fizeram parte dessa caminhada, meus professores, amigos, colegas de classe, todos que estiveram presente de alguma forma neste meu mestrado.

Obrigada Senhor, obrigada Maria, obrigada minha família (mesmo hoje sendo reduzida) e aos meus queridos Professor Bernardo e Juan.

RESUMO

Em reservatórios maduros existem centenas ou milhares de poços produtores e injetores operando simultaneamente por isso é importante conhecer o impacto dos poços injetores sobre os produtores para manter a pressão e controlar a produção de água. No caso dos reservatórios, esse mesmo impacto se dá com o intuito de evitar possíveis reativações ou fraturamento de falhas. Neste trabalho, foi proposto um workflow com duas estratégias, *reduced-physics* e *data-driven modeling*, para determinar a comunicação entre poços produtores e injetores. Ambas estratégias utilizaram dados do histórico de produção, vazões de injeção (inputs) e vazões de produção líquida (outputs). A estratégia *reduced-physics modeling* foi baseada no modelo de capacitância e resistência baseado nos produtores (CRMP) que calcula a vazão líquida do poço produtor a partir da vazão injeção e das conectividades entre injetores e produtores. As conectividades foram obtidas após minimizar o erro entre a vazão líquida observada e calculada. O algoritmo de otimização utilizado foi o de programação quadrática sucessiva (SQP). A estratégia *data-driven modeling* está baseada nas redes neurais artificiais (ANNs) que utilizou apenas dados de entrada e saída. Os parâmetros da rede neural, pesos e viéses, foram ajustados durante o processo de treinamento. Três arquiteturas foram estudadas para conectar as entradas e saídas: aprendizado simples - uma camada, aprendizado profundo – múltiplas camadas e as redes convolucionais. O algoritmo *backpropation* foi utilizado para ajustar os pesos das arquiteturas no processo de treinamento. Propomos três alternativas para o cálculo das conectividades após treinamento. A primeira baseada nos pesos ótimos. A segunda a partir do erro médio após embaralhamento dos dados de entrada e a última baseada na importância do gradiente. Dois modelos sintéticos, *Two-phases* e *Brush Canyon Outcrop*, foram utilizados para validar o workflow proposto. Os resultados mostram que as conectividades calculadas pela importância do gradiente se tornaram mais próximo das conectividades obtidas pelo modelo de capacitância e resistência quando ambos são comparados onde o erro da conectividade entre o CRMP e as outras estratégias, *Backpropagation*, *Shuffling* e CNN1D, para o injetor I-1 são 25%, 16% e 11%, respectivamente. No caso da conectividade do injetor I-3, o erro de comparar o CRMP versus *Backpropagation*, *Shuffling* e CNN1D são de 12%, 12.5% e 6%, respectivamente. Analogamente,

analisamos a conectividade do injetor I-5, obtendo os seguintes erros, 11%, 9% e 8.5%, respectivamente. Por fim, podemos concluir que a estratégia CNN1D mostra melhor aproximação no cálculo das conectividades.

Palavras-chave: aprendizado de máquina; convolutional neural networks; embaralhamento; conectividades de poços; simulação de reservatório.

ABSTRACT

In mature reservoirs, there are hundreds or thousands of producing and injecting wells operating simultaneously, so it is important to understand the impact of injection wells on producers to maintain pressure and control water production. In the case of reservoirs, this same impact aims to prevent possible reactivations or fault fracturing. In this work, we propose a workflow with two strategies, reduced-physics and data-driven modeling, to determine the communication between producing and injecting wells. Both strategies use historical production data, injection rates (inputs), and liquid production rates (outputs). The reduced-physics modeling strategy is based on the Capacitance Resistance Modeling for Producers (CRMP), which calculates the liquid flowrate of the producing well based on the injection rate, productivity index of produce, time constant and the connectivity between injectors and producers. The connectivities are obtained by minimizing the error between the observed and calculated liquid flowrates. The optimization algorithm used is the Sequential Quadratic Programming (SQP). The data-driven modeling strategy is based on Artificial Neural Networks (ANNs), which only use input and output data. The parameters of the neural network, weights, and viéseses, are adjusted during the training process. Three architectures are studied to connect the inputs and outputs: single-layer perceptron, deep learning with multiple layers, and convolutional neural networks. The backpropagation algorithm is used to adjust the weights of the architectures during training. We propose three alternatives for calculating the connectivities after training. The first one is based on the optimal weights. The second one is based on the average error after shuffling the input data, and the last one is based on the gradient importance. Two synthetic models, Two-phases and Brush Canyon Outcrop, are used to validate the proposed workflow. The results show that the connectivities calculated using gradient importance became closer to the connectivities obtained by the capacitance and resistance model when both are compared. In the case of connectivity error between CRMP and other strategies, Backpropagation, Shuffling, and CNN1D, for injector I-1, they are 25%, 16%, and 11%, respectively. Regarding the connectivity of injector, I-3, the error in comparing CRMP versus Backpropagation, Shuffling, and CNN1D is 12%, 12.5%, and

6%, respectively. Similarly, when analyzing the connectivity of injector, I-5, the errors are 11%, 9%, and 8.5%, respectively. In conclusion, we can say that the CNN1D strategy shows a better approximation in calculating connectivities.

Keywords: machine learning; convolutional neural networks; shuffling; well connectivity; reservoir simulations.

LISTA DE FIGURAS

Figura 1 - Representação esquemática do processamento dos dados de entrada e saída	25
Figura 2 - Versão simplificada das equações que descrevem as Leis de Fourier, Ohm, Darcy e Faraday	26
Figura 3 - Perceptrons: neurônio artificial e suas entradas e saída.....	32
Figura 4 - Neurônio sigmoide na rede neural	34
Figura 5 - Função Sigmoidal	35
Figura 6 - Rede Neural	35
Figura 7 - Rede neural com formulação matemática.....	36
Figura 8 - Comparação do valor do gradiente para os algoritmos de treinamento....	39
Figura 9 - plot de uma rede neural	39
Figura 10 - acurácia na rede neural de classificação	40
Figura 11 - acurácia para os dados de treinamento	41
Figura 12 - Acurácia com o aumento do número de dados.....	42
Figura 13 - Acurácia com Regularização L2.....	43
Figura 14 - estrutura dos dados de entrada e saída da rede neural.....	44
Figura 15 – Parâmetros utilizados no método do CNN1D.....	48
Figura 16 – parâmetros a serem informados pelos métodos a serem estudados	49
Figura 17 – Modelo sintético de duas fases: (a) campo de permeabilidade (b) saturação de água.....	52
Figura 18 – Modelo sintético de duas fases: Taxa de entrada para todos os injetores mudando a cada 30 dias	53
Figura 19 – Modelo sintético de duas fases: predição da produção líquida e a real para todos os produtores.....	53
Figura 20 – Modelo sintético de duas fases: produção líquida real e prevista para todos os Produtores por meio do feedforward.....	55
Figura 21 - Modelo Sintético de Duas Fases: Produção líquida prevista e real para todos os Produtores pelo método de embaralhamento.....	57
Figura 22 – Modelo sintético de duas fases: a previsão da produção líquida e a real para o produtor 1	58

Figura 23 – Modelo sintético de duas fases: a previsão da produção líquida e a real para o produtor 2.....	59
Figura 24 - Modelo sintético de duas fases: a previsão da produção líquida e a real para o produtor 3.....	60
Figura 25 - Modelo sintético de duas fases: a previsão da produção líquida e a real para o produtor 4.....	61
Figura 26 – Modelo Brush Canyon Outcrop model: Permeabilidade (a) e locação de poços (b)	64
Figura 27 – Modelo BCO: Predição da vazão líquida e vazão real pelo CRMP	65
Figura 28 – Modelo Brush Canyon Outcrop: Produção líquida prevista e real para todos os Produtores pelo método de retropropagação	67
Figura 29 – Modelo Brush Canyon Outcrop: Produção líquida prevista e real para todos os Produtores pelo método de embaralhamento.....	69
Figura 30 – Modelo Brush Canyon Outcrop: Produção líquida prevista e real para o produtor 1 e todos os injetores	70
Figura 31 – Modelo Brush Canyon Outcrop: Produção líquida prevista e real para o produtor 2 e todos os injetores	71

LISTA DE TABELAS

Tabela 1 - Parâmetros de escolha da rede neural para o caso Feedfoward	50
Tabela 2 - parâmetros de escolha da rede neural para o caso do Embaralhamento	50
Tabela 3 - parâmetros de escolha da rede neural para o caso do Gradiente	50
Tabela 4 – Modelo sintético de duas fases: Conectividades pelo CRMP.....	54
Tabela 5 – Modelo sintético de duas fases: Conectividades por feedfoward	56
Tabela 6 – Modelo Sintético de Duas Fases: Conectividade por Shuffling	62
Tabela 7 – Modelo Sintético de Duas Fases: Conectividade por CNN1D	62
Tabela 8 – Modelo Sintético de Duas Fases: Comparação de conectividades com o produtor 1 pelas diferentes técnicas.....	63
Tabela 9 – Modelo Brush Canyon Outcrop Conectividades pelo CRMP.....	66
Tabela 10 – Modelo Brush Canyon Outcrop model: Conectividade pelo feedfoward.	68
Tabela 11 - Modelo Brush Canyon Outcrop: Conectividades pelo Shuffling	72
Tabela 12 - Modelo Brush Canyon Outcrop: Conectividades pelo CNN1D.....	72

SUMÁRIO

1	INTRODUÇÃO.....	16
1.1	Objetivos Gerais	18
1.2	Objetivos Específicos.....	18
2	REVISÃO DE LITERATURA.....	19
2.1	Recuperação Secundária – Injeção de Água.....	19
2.2	Recuperação Secundária – Influencia na rocha-reservatório	20
2.3	Conectividades utilizando o modelo de capacitância e resistência.....	21
2.4	Conectividades utilizando aprendizado de máquina	22
2.5	Estrutura do Trabalho	23
3	MODELO DE CAPACITÂNCIA E RESISTÊNCIA – CRM	25
3.1	Formulação do Problema CRM.....	28
4	APRENDIZADO DE MÁQUINA	32
4.1	Perceptrons.....	32
4.2	Neurônios Sigmoides.....	34
4.3	Arquitetura de Uma Rede Neural.....	35
4.4	Tipos de Rede Neurais: Regressão e Classificação	36
4.4.1	Regressão	36
4.4.2	Classificação	37
4.5	Backpropagation ou retropropagação	38
4.6	Overfitting e Regularização	39
4.7	Aprendizado Profundo – Deep Learning.....	43
5	METODOLOGIA.....	44
5.1	Estratégia <i>Feedforward</i>	44
5.2	<i>Shuffling</i> ou Embaralhamento.....	45
5.3	Gradient Importance – Método do Gradiente.....	47
5.4	Determinação dos Parâmetros para Cada Uma das Rede Neurais.....	49
6	RESULTADOS E DISCUSSÕES.....	51
6.1	Exemplo 1: Modelo Sintético de Duas Fases.....	51
6.1.1	Caso 1: Conectividade pela Modelo de Capacitância e Resistência do Produtor - CRMP	53
6.1.2	Caso 2: Conectividade pelo Feedforward.....	55
6.1.3	Caso 3: Conectividades por Embaralhamento – <i>Shuffling</i>	57
6.1.4	Caso 4: Conectividade pelo gradiente.....	62
6.2	Exemplo 2: Modelo Brush Canyon Outcrop - (BCO).....	63

6.2.1	Caso 1: Conectividade pela Modelo de Capacitância e Resistencia do Produtor - CRMP	65
6.2.2	Caso 2: Conectividade pelo Feedfoward.....	66
6.2.3	Caso 3: Conectividade pelo Embaralhamento - <i>Shuffling</i>	68
6.2.4	Caso 4: Conectividade pelo Gradient Importance – Gradiente	72
7	CONCLUSÃO	74
	REFERÊNCIAS.....	76
	ANEXO A – ANEXO DO NOTEBOOK DO MÉTODO <i>FEEDFOWARD</i>	81
	ANEXO B – ANEXO DO NOTEBOOK DO MÉTODO DO <i>SHUFFLING</i> OU EMBARALHAMENTO	83
	ANEXO C – ANEXO DO NOTEBOOK DO MÉTODO DO GRADIENTE OU CNN1D91	

1 INTRODUÇÃO

No reservatório de óleo operam simultaneamente centenas ou milhares de poços produtores e injetores. Os poços injetores impactam no campo de maneira positiva e negativa. Positivamente, aumentando a pressão no reservatório e deslocando o fluido na direção do produtor. Negativamente pode gerar rápida irrupção de água nos produtores, fraturamento e/ou reativações de falha. Gerando uma maior informação do impacto dos injetores sobre os poços produtores podemos calcular as conectividades entre um par de poços.

Nas últimas décadas a simulação de reservatórios vem sendo empregada como uma ferramenta prática que permite gerenciar e prever a produção do campo, porém, requer conhecimento prévio das propriedades físicas e geológicas do reservatório (e.g., malha, saturação, permeabilidade, porosidade, etc.), e também resolver as equações diferenciais. No entanto, uma única simulação do reservatório pode custar horas para obter uma resposta (e.g., modelo Olympus) e consequentemente para tomar uma decisão em campo. No caso dos simuladores comerciais, cujos métodos são não-intrusivos (e.g., IMEX) e não fornecem as conectividades entre poços; alternativamente, podemos utilizar modelos proxy (*reduced-physics* ou *data-driven modeling*) baseados em dados do histórico de produção (vazões de injeção e produção) para inferir-las.

Na atualidade os modelos proxys vem sendo utilizados para descrever modelos analíticos sem resolver equações matemáticas. Na indústria do petróleo estes vem sendo utilizados para substituir os simuladores de reservatório para prever as respostas dos controles de produção (vazão de óleo e vazão de água) (Kim and Durlofsky, 2022).

O modelo de capacitância e resistência baseado nos produtores (CRMP) é um modelo matemático com física reduzida baseado nos dados do histórico da produção proposto por Sayarpour et al., (2007) para determinar as conectividades entre pares de poços injetores/produtores. O conhecimento das conectividades permite entender as propriedades do reservatório (e.g., canais de alta permeabilidade, e falhas selantes). O modelo de CRMP foi amplamente estudado em diferentes modelos

sintéticos (Kaviani et al., 2013; Mamghaderi et al., 2013; Moreno, 2013; Holanda et al., 2015; Lins et al., 2017) e campos reais (Al-Yousef et al., 2005; Weber 2009; Yang and Urdaneta, 2016; Kim, 2019; Mamghaderi et al., 2021) para determinar as conectividades entre o par de poços. Os parâmetros do CRMP podem ser obtidos resolvendo um problema de mínimos quadrados restritos.

O CRMP tem sido acoplado com outras técnicas, por exemplo, Oliveira et al. (2021) propôs uma estratégia híbrida (CRMP + Kogen) para maximizar o valor presente líquido (NPV). Tueros et al. (2018) utiliza o valor das conectividades para refinar a matriz de sensibilidade no método baseado em ensembles. Naudomsup e Lake (2018) combinam a estratégia de traçadores para melhorar a estimativa das características do reservatório. Davudov e Ventkatraman (2020) integram o CRMP com o simulador de reservatórios para melhor caracterização do reservatório. Neste trabalho, desenvolvemos e utilizamos o CRMP como um modelo de alta-fidelidade para comparações com o modelo data driven.

A rede neural artificial (ANN) é um modelo matemático capaz de reproduzir qualquer função contínua baseado em dados de entradas e saídas. As entradas são conectadas com as camadas ocultas, que por sua vez são conectados com as saídas através de pesos e vieses. Os parâmetros da rede neural são ajustados durante o processo de treinamento utilizando o algoritmo de *Backpropagation*.

Neste trabalho, utilizamos três tipos de arquiteturas (uma camada, múltiplas camadas e convolucionais) para conectar as entradas e saídas. A principal vantagem de utilizar ANNs é que a estratégia requer apenas os dados de entradas e saídas.

Nos últimos anos as ANNs vem sendo amplamente empregadas no gerenciamento de reservatório. Em exploração (Ross, 2017; Canning et al., 2017), em perfuração (Li et al., 2018; Su et al., 2021), em produção (Sun e Ertekin, 2020; Kim e Durlofsky, 2021). Por outro lado, também vem sendo utilizado para determinar a conectividade entre poços produtores e injetores. Por exemplo, Panda e Chopra (1998) utiliza para redes neurais para determinar as conectividades entre poços. Artun (2016) utiliza feedforward-ANN para quantificar as conectividades entre par de poços e determinar as falhas no reservatório. Cheng et al. (2020) utiliza análise de sensibilidade global baseado em long short-term memory (LSTM) para quantificar as

conectividades entre produtores e injetores, outros estudos relacionados podem ser encontrados em (Liu et al., 2020; Yu et al., 2020). Neste trabalho, propomos três alternativas para o cálculo das conectividades após treinamento. A primeira está baseada nos pesos ótimos. A segunda a partir do erro médio após embaralhamento dos dados de entrada e a última baseada na importância do gradiente ótimo. Finalmente, o workflow é aplicado a dois modelos sintéticos, Two-phases e Brush Canyon Outcrop para validação.

1.1 Objetivos Gerais

O objetivo deste trabalho é calcular a conectividade entre par de poços, injetores e produtores, utilizando o modelo de capacitância e resistência CRM e uma nova proposta utilizando a inteligência artificial. Logo, ambas metodologias serão aplicadas a dois modelos de reservatório.

1.2 Objetivos Específicos

Para atingir nosso objetivo geral propomos os seguintes objetivos específicos:

- Implementar o modelo de Capacitância e Resistência, baseados em poços produtores, para o cálculo das conectividades dos poços presentes no reservatório
- Utilizar métodos de inteligência artificial para calcular as conectividades entre poços injetores e produtores a partir do histórico de produção do reservatório: Utilizando redes neurais artificiais – ANN.
- Aplicar a metodologia proposta a dois modelos de reservatórios existentes na literatura.
- Avaliação e comparação dos resultados do modelo de Capacitância e Resistência com os resultados dos métodos de Inteligência Artificial.

2 REVISÃO DE LITERATURA

Nessa seção iremos apresentar os conceitos de injeção de água, bem como a importância do cuidado da não reativação de falhas nos reservatórios de petróleo. Além de revisarmos os trabalhos que utilizam o modelo de capacitância e resistência, bem como os métodos com aprendizado de máquina

2.1 Recuperação Secundária – Injeção de Água

A engenharia de reservatórios de petróleo se preocupa principalmente com a retirada de fluídos nas rochas reservatórios, se fazendo necessário o conhecimento das propriedades básicas da rocha e do fluído nela contido. Estes, determinam a quantidade do fluído no meio poroso, a capacidade de se mover no meio e principalmente a quantidade de fluído que pode ser extraído.

O petróleo é resultado do processo de degradação da matéria orgânica em subsuperfície, que acontece em uma rocha denominada rocha geradora. Posteriormente, após o processo adequado de maturação, ele é armazenado em uma rocha, denominada reservatório, a qual é selada pela rocha capeadora (rocha selante) que impede que os fluidos escapem para superfície. Os reservatórios de óleo e gás são meios porosos que podem ter qualquer origem, mas necessariamente devem apresentar características como boa porosidade e permeabilidade, bem como a capacidade de manter os fluidos em seu interior (ROSA et al., 2011).

A injeção de um fluído nos reservatórios com a finalidade de deslocar o óleo entre os poros da rocha é comumente chamado de recuperação convencional. Determinado o tipo de recuperação e o líquido a ser injetado, os projetos de engenharia de reservatórios tem a comum necessidade da implementação de poços injetores e poços produtores. Os projetos devem conter especificações como a quantidade de poços, a sua distribuição, pressões e vazões de injeções, estimativa das vazões de produção e o volume de fluídos que serão injetados e produzidos. Estes dados são importantes não só para a escolha dos equipamentos, como também para a sua viabilidade econômica. A injeção de água, nestes casos (fluído deslocam-te), é feita com o intuito de que não se misturem ou interfira na rocha-reservatório.

2.2 Recuperação Secundária – Influencia na rocha-reservatório

Um dos desafios encontrados ao utilizarmos técnicas de otimização para melhorar a performance de reservatórios com injeção de água está relacionado a determinação da influência existente entre poços injetores e produtores.

As falhas em um reservatório podem ser definidas como superfícies ou zonas estreitas onde as rochas estão cisalhadas e deslocadas, correspondendo a descontinuidades comuns em reservatórios de petróleo. Essas estruturas têm origem a partir do dinamismo litológico em conjunto com o acúmulo de tensões em subsuperfície. Enquanto selada, funciona como uma barreira impermeável para o sistema, confinando o fluido no reservatório.

A reativação de falhas geológicas é um fenômeno em que falhas são novamente ativadas devido às mudanças de pressão e tensões in situ associadas à produção e injeção de fluidos em reservatórios, como descrito por estudos como os de Guimarães et al. (2010), Rutqvist (2012), Pereira et al. (2014), Pereira (2015), Gomes et al. (2016) e Lesueur, Poulet e Veveakis (2020), Abbassi et al. (2022), entre outros. O uso da otimização de reservatórios busca trabalhar de maneira que não ocorra essa mudança brusca das falhas presentes em reservatório.

Segundo Lima (2023) O processo de reativação de falhas é caracterizado por deformações cisalhantes que reduzem a capacidade selante da falha. A dilatância, que ocorre em conjunto com o cisalhamento, leva ao aumento do volume do material componente da falha e ao surgimento de permeabilidade. Como resultado, estruturas que eram selantes passam a ser condutoras de fluidos, podendo também causar deformações superficiais, danos em tubulações, subsidência e exsudação de fluidos na superfície do fundo do mar.

Sabendo que as falhas são facilmente reativadas quando se tem a mudança de pressão in situ associadas à injeção de fluidos e da produção, conhecer os valores de injeção e sua influência sobre a produção dos poços de petróleo se torna uma das formas de evitar-se esse tipo de problema.

Diversas metodologias têm sido desenvolvidas para mensurar quantitativamente esta relação de conectividade entre poços injetores e produtores,

entre elas pode-se destacar a utilização Modelos de Capacitância e Resistencia Albertoni e Lake (2003); Youself et al. (2006); Sayarpour (2008), Lins (2018) e Oliveira (2021) bem como os métodos utilizando inteligência artificial: utilização de aprendizado de máquina, Machine learning (ML) com a utilização de Backpropagation por Artun (2016) aprendizado profundo (Yewgat et al. 2022).

2.3 Conectividades utilizando o modelo de capacitância e resistência

Albertoni e Lake, (2003), iniciaram os estudos do modelo de capacitância e resistência (Capacitance Resistance Model, CRM) que implica em um modelo simplificado de reservatório que calcula a conectividade entre poços injetores e produtores baseados no histórico das vazões de injeção e produção, que são os dados de entrada e saída, respectivamente. O modelo considera efeitos de capacitância (compressibilidade) e resistividade (transmissibilidade) e foi escolhido o nome CRM devido à analogia a um circuito elétrico resistor-capacitor.

Liang (2009) utiliza o modelo de capacitância e resistência para o cálculo das conectividades, ele faz o uso dos valores de injeção e produção líquida como dados de entrada e saída. Já Moreno (2013) emprega o CRM em um modelo de múltiplas camadas e consegue fazer a dedução das conectividades ao longo do tempo de produção.

Para o avanço desse estudo, Lins (2018) propôs dois modelos: regressão linear multivariada (Multivariate Linear Regression, MLR) e regressão linear multivariada balanceada (Balanced Multivariate Linear Regression, BMLR), onde o uso do modelo dependeu do tipo do fluxo e dos dados que estavam sendo analisados o sucesso deste modelo depende muito da disponibilidade dos dados. Ainda assim, Lins conseguiu calcular a conectividade dos reservatórios que usou de exemplo, descrevendo que obteve bons resultados.

Já Oliveira (2021) utilizou os modelos de capacitância e resistência baseado no produtor (CRMP) e os modelos de fluxo fracionário Koval e Gentil para determinar a produção de óleo e água em reservatórios com injeção de água (*Waterflooding*), acrescentando mais uma utilidade ao modelo de capacitância e resistência e suas respectivas conectividades calculadas. Oliveira também descreve que os resultados demonstram a capacidade do CRMP em prever a produção líquida dos poços e indicar

a influência dos poços injetores sobre os produtores, indicando a qualidade do método para o cálculo das conectividades.

2.4 Conectividades utilizando aprendizado de máquina

É notório que o uso da inteligência artificial vem ganhando força nos últimos meses, não diferente, no caso do cálculo das conectividades também se teve esse avanço, onde o uso do aprendizado de máquina se faz presente na área dos reservatórios de petróleo em exploração (Ross, 2017; Canning et al., 2017) em perfuração (Li et al., 2018; Su et al., 2021), em produção (Sun e Ertekin, 2020; Kim e Durlofsky, 2021) bem como na área dos dados geofísicos de reservatórios, onde no estudo de Alaei e Alaei (2013), é observado o uso de técnicas de análise de dados e redes neurais para estimar parâmetros do reservatórios, eles buscaram quantificar os parâmetros do reservatório de Marun, localizado em Ahwaz, Irã, utilizando dados geofísicos disponíveis. Para isso, os autores descreveram um método integrado que combina a Análise de Componentes Principais (*Principal Component Analysis* (PCA)), a (*Linear Discriminant Analysis* (LDA)) e redes neurais para estimar propriedades do reservatório com base em dados geofísicos. Eles argumentam que este método é superior aos individuais, pois a PCA e a LDA são usadas para selecionar as melhores variáveis de entrada para a rede neural e para reduzir a dimensionalidade dos dados. Os autores testaram o método em um conjunto de dados de reservatório de petróleo e compararam os resultados obtidos com os previstos usando apenas uma RNA ou apenas a PCA e a LDA. Eles concluíram que o método integrado produziu resultados mais precisos do que os métodos individuais.

Na determinação de falhas as aplicações associadas ao encontro e interpretação de falhas são discutidas no trabalho de Hamzeh (2021), que é uma revisão bibliográfica com o objetivo de fornecer uma atualização sobre o estado atual das técnicas de aprendizado de máquina na detecção de falhas sísmicas. Para isso, a revisão inclui a discussão sobre o uso de técnicas de ML para detecção e interpretação de falhas, revisando 79 artigos que demonstram a importância e a eficácia dessas técnicas para a interpretação de falhas. É possível notar a predominância do uso de redes neurais convolucionais para interpretação de falhas.

As pesquisas de Yuan et al. (2019), Gao et al. (2021) e Zhu et al. (2021) também se dedicam à interpretação de falhas através do uso de aprendizado de máquina.

Por outro lado, também vem sendo utilizado para determinar a conectividade entre poços produtores e injetores. Por exemplo, Panda e Chopra (1998) utiliza para redes neurais para determinar as conectividades entre poços.

Artun (2016) utiliza *feedforward*-ANN para quantificar as conectividades entre par de poços e determinar as falhas no reservatório. Cheng et al. (2020) utiliza análise de sensibilidade global baseado em *long short-term memory* (LSTM) para quantificar as conectividades entre produtores e injetores, outros estudos relacionados podem ser encontrados em (Liu et al., 2020; Yu et al., 2020). Neste trabalho, propomos três alternativas para o cálculo das conectividades após treinamento. A primeira está baseada nos pesos ótimos. A segunda a partir do erro médio após embaralhamento dos dados de entrada e a última baseada na importância do gradiente ótimo. Finalmente, o workflow é aplicado a dois modelos sintéticos, *Two-phases* e *Brush Canyon Outcrop* para validação e comparação.

2.5 Estrutura do Trabalho

Nos capítulos seguintes daremos continuidade ao trabalho, onde:

No capítulo 3 descreveremos o modelo de capacitância e resistência, por quem foi originalmente elaborado e como sua formulação matemática foi feita para ser utilizada em poços de reservatório.

No capítulo 4 teremos a introdução ao aprendizado de máquina, bem como sua revisão, onde serão inseridos os tópicos e assuntos principais que envolvem as redes neurais e seus componentes e tipos

No capítulo 5 será então feito a nossa metodologia ou necessariamente as estratégias utilizadas no decorrer da pesquisa. As estratégias que foram utilizadas e como foram utilizadas até chegarmos aos resultados destas.

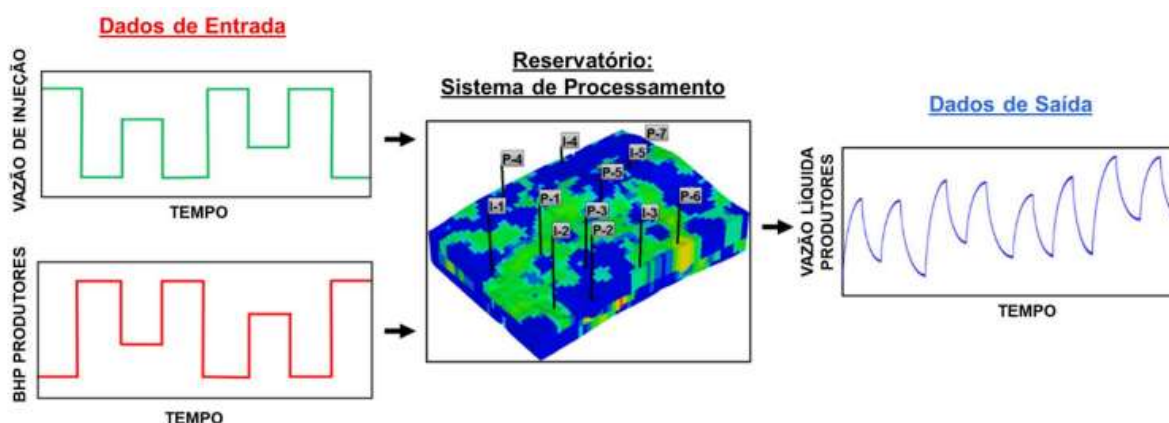
No capítulo 6 teremos os resultados das nossas estratégias. Consequentemente no capítulo 7 teremos a conclusão e em sequência a revisão bibliográfica.

3 MODELO DE CAPACITÂNCIA E RESISTÊNCIA – CRM

O modelo de capacitância e resistência (CRM), é um modelo simplificado de reservatório que quantifica a conectividade entre poços injetores e produtores, baseado no histórico das vazões de injeção e produção, os quais são tratados como dados de entrada e saída, respectivamente, e o reservatório um sistema que processa dados. Esses dados estão disponíveis de forma abundante nos campos e através deles podemos determinar os parâmetros do CRM, que são as conectividades, constante de tempo e índice de produtividade, sem que seja necessária a modelagem física do reservatório. Tais parâmetros refletem a dinâmica do fluxo dos fluidos nos reservatórios assim como os efeitos das heterogeneidades (canais e falhas) existentes no mesmo.

O modelo foi desenvolvido inicialmente por Albertoni e Lake (2003) e tem como base a relação entre os efeitos de capacitância (compressibilidade) e resistividade (transmissibilidade) de um dado meio poroso. Bruce (1943) foi o primeiro a apresentar um modelo análogo experimental entre o fluxo de óleo em meio poroso e o fluxo de corrente elétrica em meio condutor. Seus experimentos permitiram desenvolver uma unidade de aparatos elétricos que representavam fisicamente uma malha computacional de um reservatório, ao relacionar as equações que governam o fluxo em meios porosos com as que descrevem os circuitos elétricos. Assim como ilustrado na Figura 1 abaixo:

Figura 1 - Representação esquemática do processamento dos dados de entrada e saída



Fonte: Oliveira et. al (2019)

Oliveira et. al (2021) por meio de Bruce (1943) mostra as equações mais importantes da física, observando a Figura 2, é possível compreender melhor a estruturação do modelo de capacitância e resistência. Todas essas equações têm em comum uma relação de causa e efeito, há uma força motriz (Diferença de Potencial, Tensão, Pressão ou Temperatura) que gera um fluxo (elétrico, de fluidos, magnético ou calor) resultante de sua aplicação.

Figura 2 - Versão simplificada das equações que descrevem as Leis de Fourier, Ohm, Darcy e Faraday

$q_c = -kA \frac{\partial T}{\partial x}$	<i>Lei de Fourier</i>
$i = R \cdot \frac{\partial V}{\partial t}$	<i>Lei de Ohm</i>
$q_f = -KA \cdot \frac{\partial P}{\partial x}$	<i>Lei de Darcy</i>
$\varepsilon = -N \cdot \frac{\partial \Phi_B}{\partial t}$	<i>Lei de Faraday</i>

■ CAUSA ■ EFEITO

Fonte: Oliveira et. al (2019)

Ainda descrito, da mesma forma que um diferencial de pressão gera um fluxo de fluidos em um meio poroso, um diferencial de tensão produz uma corrente elétrica (fluxo de elétrons) num circuito elétrico. Ambos os fluxos são influenciados por características intrínsecas ao meio como pela intensidade das forças motrizes as quais estão submetidos. Analogamente, um reservatório de hidrocarbonetos apresenta também propriedades capacitivas as quais são demonstradas em termos da compressibilidade das rochas e dos fluidos, as quais permitem que energia seja armazenada na forma de pressão. Por outro lado, o meio poroso também demonstra resistência a passagem de fluidos, e esta é medida pelo inverso da transmissibilidade que ele possui.

Assim, baseando-se na relação de capacitância e resistência que os meios condutores apresentam a passagem da corrente elétrica, Yousef et al. (2005) propôs o primeiro modelo matemático, que posteriormente seria denominado Modelo de Capacitância e Resistência (CRM), aplicado a reservatórios de petróleo. Entretanto, foram Albertoni e Lake (2003) os responsáveis por quantificar as relações entre os poços e sua relação com os dados de produção, como por introduzir os principais parâmetros do modelo (conectividades e constante de tempo) e a estrutura matemática base para estudos posteriores. Lins et al. (2017) descreve que eles utilizaram filtros difusivos para considerar o tempo de atraso e atenuação das mudanças que ocorrem entre o estímulo (injeção) e a resposta (produção). Os filtros se tornam importantes quando existem grandes distâncias entre injetores e produtores e para meios com grande dissipação do fluido. A dissipação é definida como o inverso da difusividade. Baixas permeabilidades, grande porosidade e uma compressibilidade total alta, contribuem para uma grande dissipação. Se não houver dissipação no reservatório, e dependendo da distância entre os poços produtor e injetor, uma mudança na vazão de injeção causa uma mudança equivalente e instantânea na vazão de produção. Os filtros são aplicados às vazões de injeção e basicamente transformam essas vazões de forma que quando influenciarem a vazão de um determinado produtor a resposta deste último será equivalente à de um meio incompressível, ou seja, o efeito da dissipação será anulado. Para a utilização dessa metodologia, os autores consideraram algumas hipóteses, como:

Para a derivação do modelo CRM, são adotadas as seguintes premissas:

- O reservatório é homogêneo;
- Não há alterações nas configurações geológicas do reservatório, o qual permanece inalterado durante toda janela de tempo estudada;
- O número de poços é considerado constante;
- Os parâmetros operacionais do campo e as propriedades da rocha reservatório devem ser constantes, com exceção das vazões de injeção, produção e pressão de fundo de poço (BHP) dos poços produtores e injetores

3.1 Formulação do Problema CRM

O CRMP ou Modelo de Capacitância e Resistência Baseado no Produtor, é um modelo que representa o reservatório em volumes de controle definidos pelas zonas de drenagem de cada poço produtor, que incluem todos os poços injetores que influenciam sua vazão de produção.

O CRMP considera três parâmetros principais: a constante de tempo τ_j , o índice de produtividade (J_j) e as conectividades ($\lambda_{i,j}$). Os dois primeiros são determinados para cada poço produtor j , enquanto as conectividades são determinadas para cada par injetor i – produtor j .

A descrição dos parâmetros do modelo CRMP é apresentada a seguir:

- Conectividades (λ_{ij}): São definidas fisicamente como a fração do volume injetado pelo poço injetor i que influencia a produção de um dado poço produtor j , indicando o volume de água que escoar em direção ao mesmo. Além disso, ela reflete a transmissibilidade entre os poços e, segundo Yousef et al. (2005), pode ser entendida como uma propriedade que se relaciona diretamente as características geológicas do reservatório sendo também influenciada pela posição dos poços.
- Constante de Tempo (τ_j): Refere-se ao tempo necessário para que uma variação no volume injetado influencie a vazão de produção de um dado poço produtor j , através da variação da pressão no meio poroso (Yousef et al., 2005). Se o τ_j possui um valor muito alto significa que o sinal (a onda de variação de pressão devido a injeção de fluidos no reservatório) demora a se propagar no meio poroso, ou seja, o sistema apresenta uma resposta lenta ao estímulo. Se é muito pequeno, os efeitos inversos são observados.
- Índice de Produtividade (J_j): Quantifica qual a variação de pressão necessária no volume poroso drenado para que se obtenha uma dada vazão $q_j(t)$

Assim, temos que a vazão líquida de um poço produtor em condições de reservatório pelo CRMP é derivada da equação de continuidade (Sayarpour et al., 2008) e pode ser expressa matematicamente

$$q_{j,k} = q_{j,k-1} e^{-\Delta t/\tau_j} + \left(1 - e^{-\Delta t/\tau_j}\right) \sum_{i=1}^{N_i} \lambda_{i,j} I_{i,k} - \left(1 - e^{-\Delta t/\tau_j}\right) \left[\tau_j J_j \frac{P_{wf,j}^k - P_{wf,j}^{k-1}}{\Delta t} \right] \quad (1)$$

Onde:

$q_{j,k}$: produção líquida total do produtor j no tempo k ;

$I_{i,k}$: valor de injeção i no tempo k i

$P_{wf,j}^k$: BHP do produtor j ;

$\lambda_{i,j}$: conectividade do par de poços injetor/produtor

τ_j : constante de tempo para o produtor j

N_i : número total de poços

J_j : índice de produtividade do produtor j .

A Equação (1) nos permite interpretar a vazão líquida total de um poço produtor j como resultado da soma de três efeitos principais (termos a direita da equação), onde o primeiro termo se refere a resposta inicial do reservatório as condições operacionais impostas, representando a fração associada a energia natural do reservatório (produção primária) sobre a produção. O segundo termo traz os efeitos da influência da recuperação secundária sobre a produção (injeção de água); e o terceiro termo representa a fração de óleo produzida decorrente dos efeitos da mudança da pressão de fundo de poço (BHP) dos produtores, podendo ser anulado caso os dados de BHP não estejam disponíveis. Neste último caso, ao desconsiderar o BHP, o parâmetro J_j também não será avaliado, reduzindo assim o número de parâmetros calculados.

Para este estudo, a determinação dos parâmetros τ_j , J_j e λ_{ij} , $i = 1 \dots N_i$, $j = 1 \dots N_j$, é feita através de um problema de otimização utilizando a técnica dos mínimos quadrados, o qual é dividido em duas partes: um problema acoplado e outro desacoplado. Essa estratégia é apresentada por Lins et al. (2017) e tem por objetivo resolver inicialmente um problema mais simples (desacoplado), onde o processo de otimização é feito de forma independente para cada produtor. Seu resultado gera um ponto de partida, warm start, para o problema acoplado que, por sua vez, realiza uma

otimização que engloba todos os produtores, sendo mais complexo. Como demonstrado por Lins et al. (2017), a grande vantagem dessa estratégia é a diminuição da quantidade de iterações do problema acoplado, resolvido através da técnica Programação Quadrática Sequencial (SQP). Dessa forma, o uso de um warm start para o problema acoplado garante uma melhor performance para a otimização. Mais detalhes podem ser encontrados em Lins et al. (2017), Lins (2018) e Tueros et al. (2018). O problema de otimização a ser resolvido, que se trata de um problema de mínimos quadrados cuja função objetivo é o somatório do quadrado dos erros entre a vazão observada ($q_j^{k,obs}$) e a calculada pelo modelo CRMP ($q_j^{k,cal}$) em cada passo de tempo k , para todos os poços produtores ativos no campo (onde $j = 1 \dots N_j$, $i = 1 \dots N_i$), tem a seguinte forma:

$$\begin{aligned}
 & \underset{\lambda_{ij}, \tau_j, J_j}{\text{Minimize}} \quad \sum_{k=1}^{N_t} \sum_{j=1}^{N_p} (q_j^{k,obs} - q_j^{k,cal})^2 \\
 & \text{sujeito a:} \quad \sum_{j=1}^{N_p} \lambda_{ij} \leq 1, \text{ para todo } i \\
 & \quad 0 \leq \lambda_{ij} \leq 1, \text{ para todo } i \text{ e } j \\
 & \quad \tau_{min} \leq \tau_j \leq \tau_{max} \\
 & \quad J_{min} \leq J_j \leq J_{max}
 \end{aligned} \tag{2}$$

Os seguintes passos são adotados para resolução do problema apresentado na Equação (2):

1. No simulador, valores aleatórios de BHP para os produtores e de vazão de injeção para os injetores são impostos dentro de um intervalo determinado. Os valores são definidos em ciclos de controle de 30 dias (LINS et al., 2017; TUEROS et al., 2018).

2. Os dados da janela de tempo analisada são definidos a partir do passo de tempo em que todos os poços produtores já atingiram a irrupção de água (Waterbreakthrough).

3. O método dos mínimos quadrados é utilizado para resolver o problema desacoplado, onde a restrição presente na Equação (2) é desconsiderada, e a otimização é feita de forma independente para cada poço produtor. A solução deste problema é o ponto de partida para resolver o problema acoplado. O problema desacoplado é dado por:

$$\begin{aligned}
 & \underset{\lambda_{ij}, \tau_j, J_j}{\text{Minimize}} \sum_{k=1}^{N_t} \sum_{j=1}^{N_p} (q_j^{k,obs} - q_j^{k,cal})^2 \\
 & 0 \leq \lambda_{ij} \leq 1, \text{ para todo } i \text{ e } j \\
 & \tau_{min} \leq \tau_j \leq \tau_{max} \\
 & J_{min} \leq J_j \leq J_{max}
 \end{aligned} \tag{3}$$

4. Para cada par produtor-injetor, é verificado se os resultados das conectividades ($\sum_{j=1}^{N_p} \lambda_{ij} \leq 1$) infringem a restrição do problema, para que o ponto de partida seja um ponto viável, Lins et al. (2017).

5. A partir do ponto de partida obtido, se resolve o problema acoplado apresentada na Equação (2), utilizando o algoritmo SQP.

Os dados para $q_j^{k,obs}$ podem ser obtidos a partir do histórico de produção do campo, porém neste trabalho, a vazão líquida observada foi gerada pelo simulador comercial CMG-IMEX (CMG, 2016). Assim, as variáveis de projeto são determinadas ao calibrar-se a vazão líquida calculada à observada. Ressalta-se que para cada poço produtor existem 3 variáveis de projeto associadas, além da q_j^{k-1} que pode estar disponível ou não, podendo também ser tratada como outra variável de projeto. Para os casos estudados, as vazões iniciais dos produtores eram conhecidas e o histórico de seus BHP constantes.

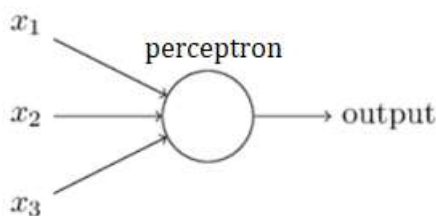
4 APRENDIZADO DE MÁQUINA

O aprendizado de máquina ou *Machine Learning* na sua tradução para o inglês é um método de análise de dados que automatiza a construção de modelos analíticos. É um ramo da inteligência artificial baseado na ideia de sistemas de aprendizado com dados, onde reconhece padrões sem necessidade de interferência humana. Aprendendo com os seus dados e seus erros, estes usam algoritmos para a previsão de dados.

4.1 Perceptrons

A primeira proposta de neurônios artificiais foi feita por McCulloch e Pitts (1943), para compreender o processamento de sinais no cérebro, sendo posteriormente aprimorado por outros, incluindo Frank Rosenblatt, que desenvolveu um neurônio artificial chamado Perceptron por volta dos anos 1950 a 1960. Com auxílio da Figura 3 abaixo, será melhor explicado como funciona um neurônio artificial.

Figura 3 - Perceptrons: neurônio artificial e suas entradas e saída



Fonte: Nielsen (2019)

O perceptron pega diversas entradas (*inputs*) x_1 , x_2 , x_3 e produz uma única saída (*output*), como na Figura 3. No exemplo temos 3 entradas, porém geralmente podemos ter muitas outras. Rosenblatt propôs uma única regra para computar a saída, ele introduziu o conceito de pesos (*weights*) w_1 , w_2 , w_3 números reais que expressam a importância da respectiva entrada para sua saída.

O dado de saída do neurônio, 0 ou 1, é determinado então pelo valor da soma dos pesos ser mais ou menos próximo do valor de referência, que é um valor real que é determinado por um parâmetro do neurônio.

Nielsen (2019) propõe um exemplo simples mostrando como os pesos funcionam no caso dos perceptrons. Ir ou não ir a um festival de queijo que haverá na

cidade; você decidirá então se irá ou não a esse festival dependendo de três condições?

O clima está bom?

Seu companheiro quer ir com você?

O transporte público é bom próximo ao festival?

representando então estas condições em inputs, temos:

x_1 : O clima está bom?

x_2 : Seu companheiro vai com você?

x_3 : O transporte público é bom próximo ao festival?

onde $x_1 = 1$ se a resposta for sim e 0 se a resposta for não, e assim sucessivamente para as outras condições.

Agora, considerando que você quer tanto ir ao festival que não importa de ir sozinho, logo, adicionaremos um peso w_2 para a entrada x_2 , já que é um parâmetro não tão importante, $w_2 = 2$. Considerando então que o clima é muito importante e te impediria de ir ao festival, colocaremos um peso $w_1 = 6$ e $w_3 = 2$. Pode-se notar que o parâmetro de maior importância é o de maior peso, w_1 , considerando que você colocou um valor referência de 5 para o perceptrons, com essas escolhas, teremos um valor de saída 1 quando o clima for bom, independente das outras condições e um valor de saída 0, quando o clima não for bom também independente das outras condições. Temos então que, variando os pesos e o valor de referência, nós temos diferentes tipos de modelos de tomada de decisões. Se mudarmos o valor referência para 3, as outras duas condições contribuirão para a tomada de decisão.

Matematicamente, podemos descrever que um perceptrons funciona da seguinte maneira:

$$output = \begin{cases} 0 & \text{se } w.x + b \leq 0 \\ 1 & \text{se } w.x + b > 0 \end{cases}$$

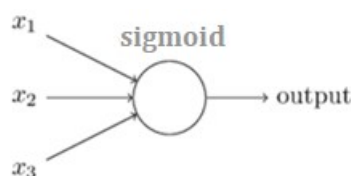
Onde, b é a viéses ou viés, que implica no valor de referência. Descrevendo o viés melhor, podemos dizer que é uma maneira de quanto mais fácil é de se fazer o

perceptrons chegar a dar uma saída no valor de 1, ou em termos biológicos, do neurônio *disparar*. Quando temos um viés muito grande, é muito fácil pra o perceptrons disparar, já quando o contrário acontece, é mais fácil que a saída seja 0. Todos esses exemplos e explicações mostram como os perceptrons conseguem resolver equações lógicas, a rede neural, formada pelos neurônios artificiais, pode facilmente aprender a resolver problemas.

4.2 Neurônios Sigmoides

Assim como os perceptrons, o neurônio sigmoide é um tipo de neurônio artificial que compõe a rede neural, eles são modificados para que qualquer mudança pequena nos pesos e vieses cause apenas uma pequena mudança na saída. Esse é o fator crucial que permite que a rede neural composta por neurônios sigmoide aprenda. No caso do sigmoide, ele aceita entradas com valores entre 0 e 1. Tem um peso para cada entrada e o viés, e sua saída não é só 0 ou 1 mas sim o valor $\sigma(w.x + b)$, onde σ é chamada função sigmoide e é descrita na equação 4 abaixo, também podemos ver a estrutura da rede neural na Figura 4 com o neurônio sigmoide e suas respectivas saídas e entradas.

Figura 4 - Neurônio sigmoide na rede neural



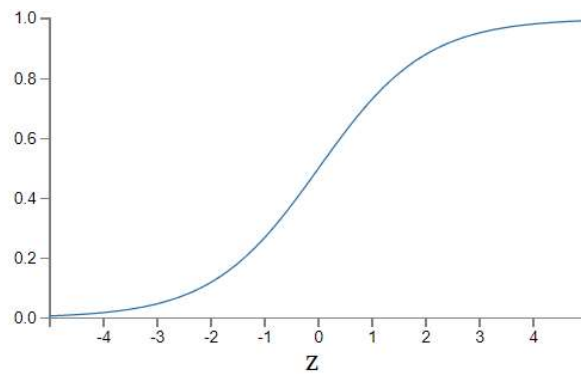
Fonte: Nielsen (2019)

Tem um peso para cada entrada e o viés, e sua saída não é só 0 ou 1 mas sim:

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (4)$$

Assim como o perceptrons, quando temos um valor $z = w.x + b$ muito grande, temos que a função e^{-z} tenderá a 0, logo $\sigma(z) = 1$, assim como o oposto, quando o valor de z for um número negativo muito grande e^{-z} tenderá a infinito, logo teremos um valor próximo de 0. Temos então o comportamento da função sigmoide na Figura 5.

Figura 5 - Função Sigmoid

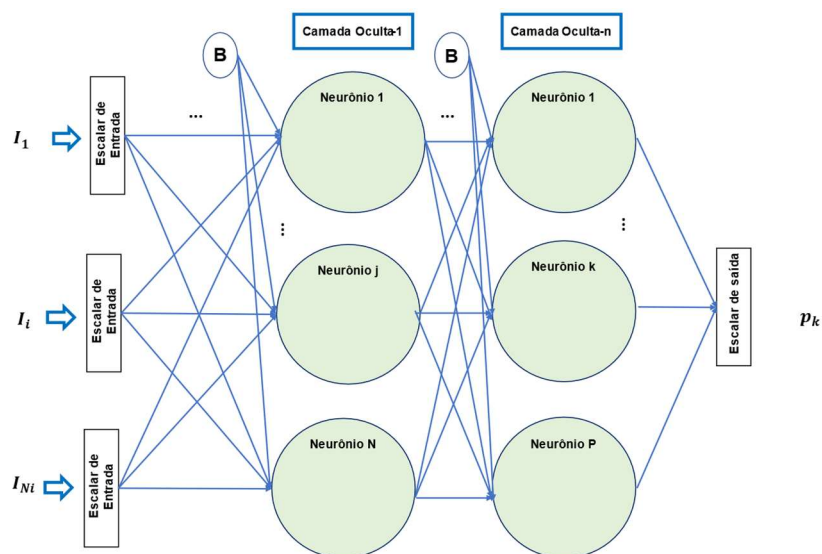


Fonte: Nielsen (2019)

4.3 Arquitetura de Uma Rede Neural

Considerando a Figura 6 abaixo, temos então de forma simples, a arquitetura de uma rede neural com duas camadas ocultas.

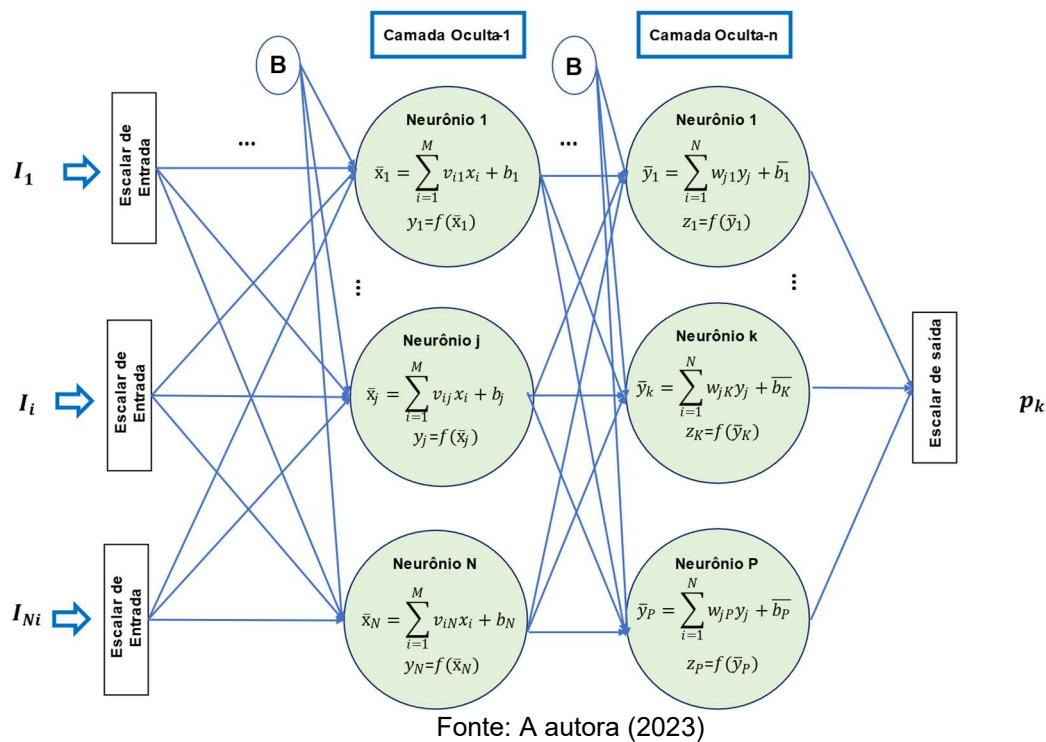
Figura 6 - Rede Neural



Fonte: A autora (2023)

Agora, considerando a formulação matemática da rede neural, temos a Figura 7.

Figura 7 - Rede neural com formulação matemática



4.4 Tipos de Rede Neurais: Regressão e Classificação

Redes neurais podem ser divididas em redes de regressão e classificação, no primeiro caso, usamos estas para previsão de dados numéricos ou não, já no segundo para a classificação.

4.4.1 Regressão

A regressão é utilizada quando queremos prever algum tipo de dado, aqui também temos a função custo, ou a função de perda ou objetivo que permite o ajuste dos pesos e vieses através da minimização desta função

$$C(W) = \sum (y_i - t_i)^2 \quad (5)$$

A equação 5 representa a função de custo, onde t_i é o vetor de valores alvo e y_i consiste no vetor de predições/saídas da rede neural. Assim, a programação direta de x fornece y , através da rede neural. A equação introduz o algoritmo de retropropagação, que calcula $l_{-1} = f(z_l)$ e então conecta a última camada oculta como uma entrada.

4.4.2 Classificação

No caso da classificação, nós temos dois tipos: a binária e a não binária. A primeira, corresponde a rede neural que deve escolher entre dois tipos de opção: sim ou não, verdadeiro ou falso, correto ou não correto, etc.

Já quando queremos mais prever mais de uma resposta, precisaremos de mais de um neurônio de saída.

Aqui nós temos a acurácia que é a métrica de desempenho mais básica para modelos de classificação, que mede a fração de pontos de dados corretamente classificados pelo modelo. Em um classificador de duas classes, a acurácia é expressa da seguinte maneira:

$$Acurácia = \frac{VP + VN}{VP + FP + VN + FN} \quad (6)$$

Onde:

VP: Verdadeiro positivo

VN: Verdadeiro Negativo

FP: Falso Positivo

FN: Falso Negativo

A acurácia deve ser usada com cautela, pois só funciona bem quando as classes estão aproximadamente balanceadas. Portanto, é importante ter cuidado ao utilizar a acurácia como medida de desempenho em problemas de classificação em que as classes podem estar desbalanceadas. Nestes casos, é recomendado utilizar outras

métricas, como a sensibilidade (recall) ou a F1-score, que são mais robustas ao desbalanceamento de classes, para resolver este problema.

4.5 Backpropation ou retropropagação

A retropropagação é o método mais comum utilizado para o treinamento de uma rede neural, foi introduzido por Rumelhart, Hinton e Williams e é utilizado até hoje.

Esse é o meio primário para determinar os valores dos pesos de uma rede neural durante seu treinamento. Funciona calculando o valor modificado do peso (v_t) para cada peso (θ) na rede neural. Esse valor é dado pela seguinte equação:

$$\theta_t = \theta_{t-1} - v_t \quad (7)$$

Repetimos o processo para cada iteração (t). O algoritmo de treinamento determina como calculamos a mudança do peso. A retropropagação clássica calcula o gradiente (∇) para cada peso na rede neural para cada função erro (J). Escala-se o gradiente por uma taxa de aprendizado (n , η)

$$v_t = n \nabla_{\theta-t} J(\theta_{t-1}) \quad (8)$$

A taxa de aprendizado é um importante conceito da retropropagação, determinar a taxa de aprendizado é complexo, pois:

Uma taxa muito baixa vai convergir em uma solução razoável, porém o processo será prologando

Uma taxa de aprendizado muito alta vai fazer com que o resultado seja convergido para um erro ainda maior do que uma boa taxa de aprendizado.

Geralmente os valores utilizados nas taxas de aprendizado são de: 0,1, 0,001, etc. Neste trabalho, também testamos valores de taxa de aprendizado para obter melhores resultados e os testamos.

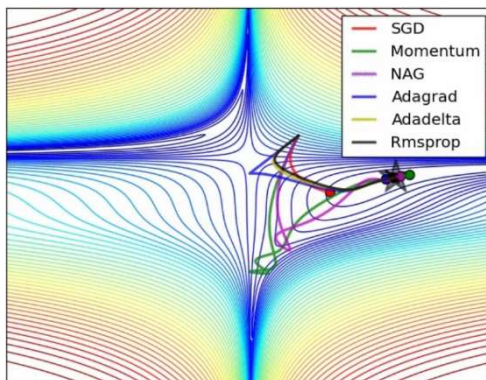
A retropropagação é um tipo de gradiente descendente, o gradiente descendente significa o cálculo do gradiente para cada um dos pesos da rede neural de treinamento. O gradiente é derivativo da função erro do valor do peso utilizado. A função erro ou função do custo mede a diferença do valor entre o dado pela rede, do valor esperado que a rede resultasse, pois trata-se de dados de treinamento que já

estão com seus valores presumidos. Utilizando o gradiente descendente, conseguimos valores ainda menores da função erro.

Tomando como base a retropropagação, temos outros tipos de algoritmos de treinamento, que também foram testados em nossa pesquisa, utilizando aquele que obtivesse melhor custo computacional e tempo.

A Figura 8 mostra diferentes algoritmos de retropropagação (treinamento) e como eles treinam ao longo do tempo.

Figura 8 - Comparação do valor do gradiente para os algoritmos de treinamento

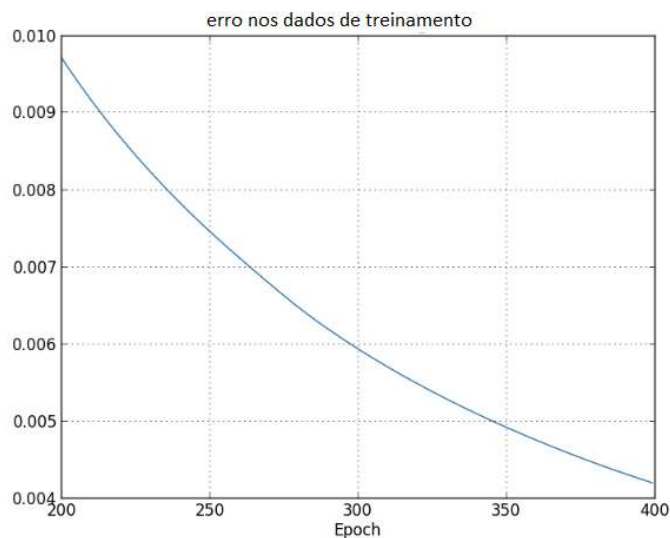


Fonte: Alec Radford (2023)

4.6 Overfitting e Regularização

Para entendermos melhor os conceitos de *overfitting*, precisamos ver um exemplo de uma rede neural, onde a função é diminuir a função erro, ou custo. Na Figura 9, podemos ver um plot de uma rede neural dividida por épocas, onde o valor da função custo vai diminuindo ao decorrer do aumento do número de dados.

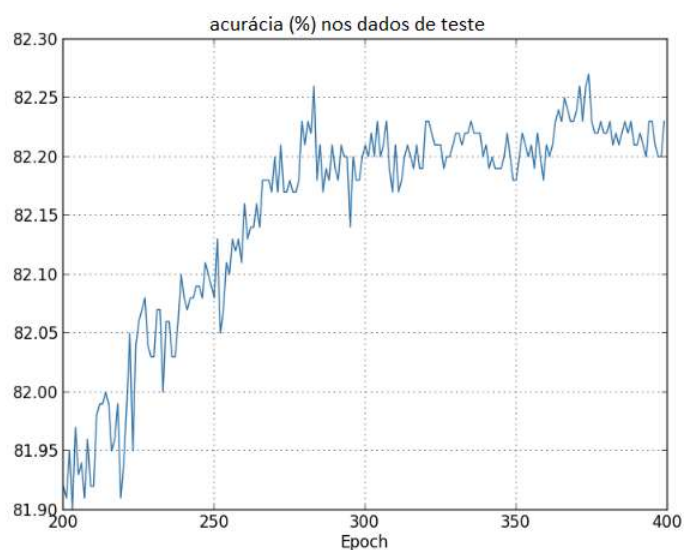
Figura 9 - plot de uma rede neural



Fonte: Nielsen (2019)

Agora olhando a acurácia de uma classificação nos dados de testes muda ao longo do tempo, ilustrado na Figura 10:

Figura 10 - acurácia na rede neural de classificação



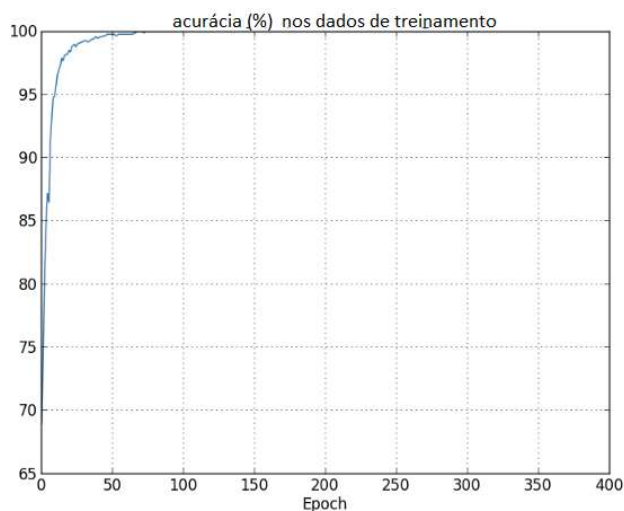
Fonte: Nielsen (2019)

Pode-se notar que a acurácia atinge o ponto máximo por volta de 82%; o aprendizado então diminui gradualmente, então até a época 280 a acurácia praticamente para de melhorar. Quando olhamos para a Figura 8, acredita-se que o treinamento continua melhorando, porém quando vemos a acurácia percebemos que

não, mantendo o valor maior de acurácia para aqueles dados de teste no valor máximo de 82%. A partir da época 280 então, o que a rede neural aprende não mais influência mais no aprendizado. Chamamos esse evento então de *overfitting* depois da época 280.

Na Figura 11, temos a demonstração da acurácia para os dados de treinamento, onde podemos ver que este atingiu 100%, a rede neural então passa a generalizar e memorizar os dados de treinamento e não a aprender com os dados, fazendo com que na hora dos dados testes, tenhamos uma acurácia muito baixa.

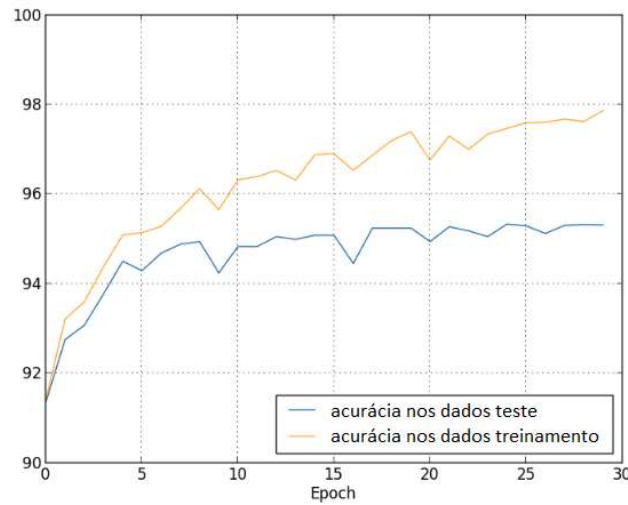
Figura 11 - acurácia para os dados de treinamento



Fonte: Nielsen (2019)

Utilizando agora um maior número de dados, conseguimos então diminuir o intervalo entre as curvas da acurácia dos dados de teste e de treinamento, como na Figura 12.

Figura 12 - Acurácia com o aumento do número de dados



Fonte: Nielsen (2019)

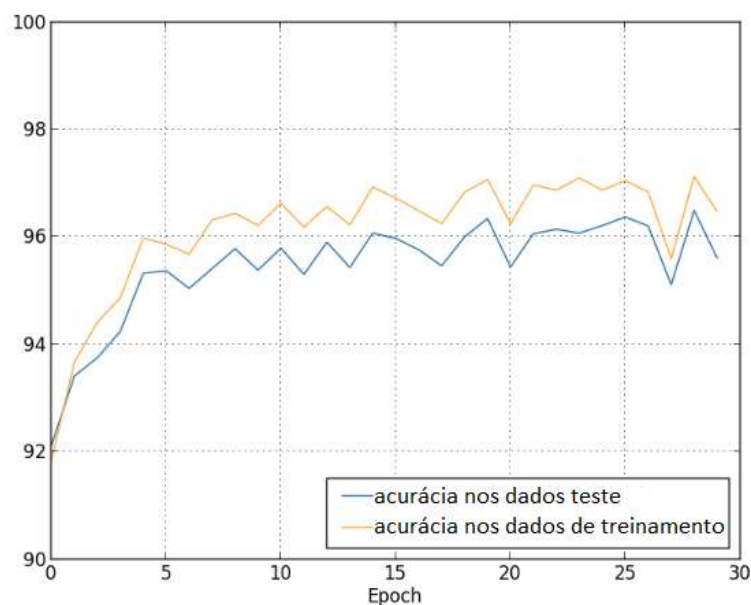
Mas esta não é a única maneira de evitar ou combater o *overfitting*, temos também a regularização. A regularização funciona adicionando um termo extra a função custo, ou função objetivo. Neste momento vamos citar a regularização L2, ou *weight decay*. A equação 9 mostra o efeito da regularização na função custo.

$$\mathcal{C} = \frac{1}{n} \sum_{x_j} y_j \ln a_j^L + (1 - y_j) \ln(1 - a_j^L) + \frac{\lambda}{2n} \sum_w w^2 \quad (9)$$

Onde o primeiro termo é a função da entropia cruzada com a adição de um segundo termo, basicamente a soma do quadrado dos pesos, o termo $\lambda/2n$ é conhecido como parâmetro regularizador e o n é usualmente o tamanho do nosso treinamento.

No mesmo caso da rede neural acima, com o mesmo número de dados, temos então a seguinte acurácia, como ilustrado na Figura 13 abaixo. Podemos notar então que o intervalo entre as curvas diminuiu significativamente após a introdução da regularização L2 na rede neural, evitando o *overfitting*.

Figura 13 - Acurácia com Regularização L2



Fonte: Nielsen (2019)

4.7 Aprendizado Profundo – Deep Learning

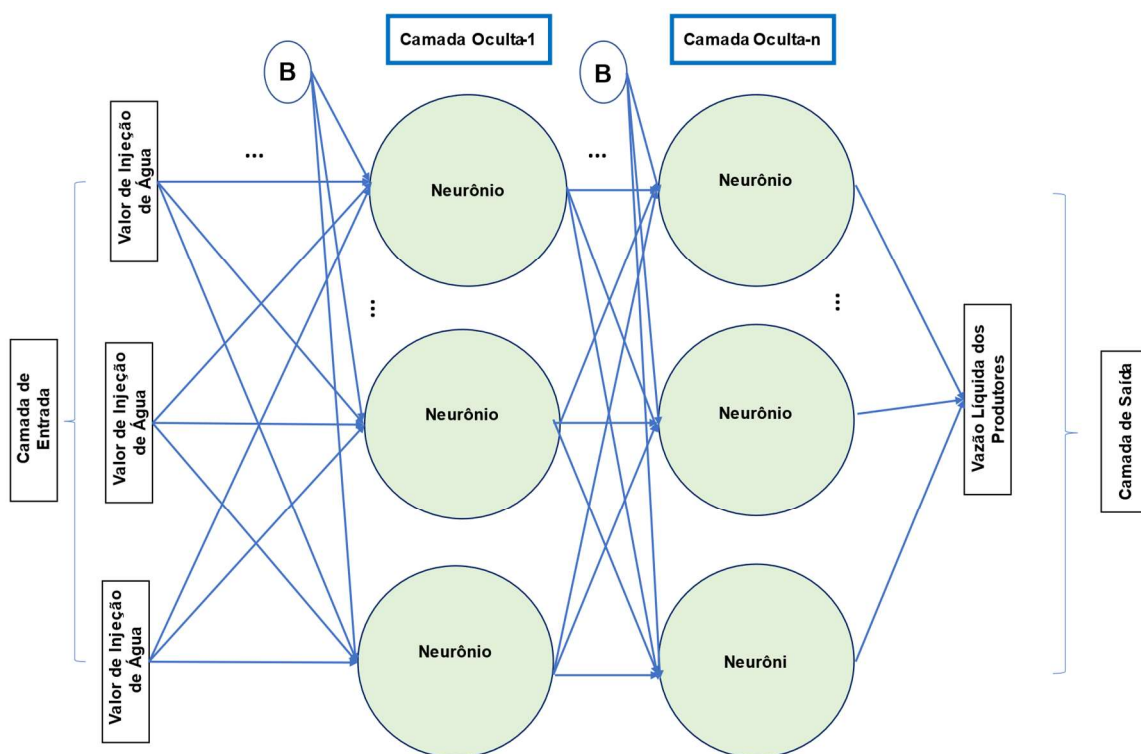
O Deep Learning, ou aprendizado profundo, nada mais é do que uma rede neural com mais de uma camada oculta. Estas redes são mais difíceis de treinar pois podem apresentar inúmeras camadas ocultas. Elas são comumente usadas e com sucesso para reconhecimento de imagens, reconhecimento de voz e outras aplicações. As redes convolucionais usam três ideias básicas: *local receptive Fields*, pesos compartilhados e o *pooling*.

5 METODOLOGIA

5.1 Estratégia *Feedforward*

A estratégia *feed Forward Backpropagation* possui apenas uma camada intermediária entre as entradas e saídas, aqui são considerados os dados de entrada e saída: vazão de injeção e a produção líquida dos produtores, como ilustrado na Figura 14.

Figura 14 - estrutura dos dados de entrada e saída da rede neural



Fonte: A autora (2023)

Observando os pesos das conexões da rede neural de treinamento conseguimos quantificar os valores das conectividades entre os poços. Os passos para calcular os pesos ótimos são:

1. Informe valores iniciais aleatórios dos pesos e vieses para conectar a arquitetura da rede (inputs, neurônios e outputs).
2. Calcular a função de ativação em cada neurônio.

3. Calcular o erro entre os valores observados e calculados da vazão líquida dos produtores.
4. Atualizar os pesos e viéses baseado no erro.
5. Repetir os passos de 2 a 4 até que a rede neural tenha atingido um nível aceitável de precisão ou critério de parada.

Dos pesos ótimos obtidos após treinamento podemos calcular as conectividades utilizando a seguinte expressão matemática proposta por Olden and Jackson, (2000):

$$\lambda_{ij}^{ann} = \frac{c_{ij}}{\sum_{j=1}^{N_p} |c_{ij}|} \text{ onde } c_{ij} = \left| \sum_{l=1}^{N_i} v_{l,r} * w_{r,k} \right| e \quad (10)$$

$r = 1 \cdots N_n, \text{ para todo } k.$

onde, $v_{i,r}$ denota o i –ésimo peso e r –ésimo neurônio, $w_{r,k}$ representa o r –ésimo neurônio e k –ésima saída. λ_{ij}^{ann} representa a conectividade entre o i –ésimo injetor e k –ésimo produtor.

5.2 *Shuffling* ou Embaralhamento

A estratégia de embaralhamento para determinar a comunicação entre par de poços está baseada no aprendizado profundo e embaralhamento dos dados de entrada.

O aprendizado profundo é uma estratégia capaz de aprender a realizar tarefas complexas com grande quantidade de dados e é composto por várias camadas ocultas. Em nosso caso a estratégia é utilizada para obter os parâmetros do modelo:

Informe valores iniciais dos pesos e viéses para conectar a arquitetura da rede (inputs, neurônios e outputs).

Calcular a função de ativação em cada neurônio das camadas ocultas.

Determinar o erro entre os valores observados e calculados da vazão líquida dos produtores.

Atualizar os pesos e vieses baseado no erro da função objetivo.

Repetir os passos de 2 a 4 até que o aprendizado profundo tenha atingido os pesos ótimos.

A estratégia de embaralhamento normalmente vem sendo utilizada para melhorar o desempenho e a generalização dos modelos de aprendizado automatizado. Além dessa finalidade, nós o utilizamos para calcular as conectividades baseado no impacto das entradas sobre as saídas após processo de treinamento:

Embaralhar todas as entradas simultaneamente, isto é, a vazão de injeção de todos os poços injetores.

Escolher o i – poço injetor do dataset original e o substituímos pelos i – poço injetor do conjunto de embaralhamento.

Do novo *dataset* e os pesos ótimos realizamos uma avaliação feedforward para calcular a vazão líquida do j – produtor baseado no embaralhamento, $q_j^{shuffled}$ com $j = 1 \dots N_p$.

Determinar o erro quadrático médio (MSE) entre a vazão líquida observada, $q_{j,k}^{obs}$, e vazão líquida calculada, $q_{j,k}^{shuffled}$, para o k -ésimo passo de tempo, baseado no impacto do i – poço injetor após embaralhamento, matematicamente pode ser expresso:

$$MSE_{ij} = \frac{1}{N} \sum_{k=1}^N (q_{j,k}^{obs} - q_{j,k}^{shuffled})^2 \text{ com } i = 1 \dots N_i.$$

Após cálculo do MSE entre todos, definimos a conectividade entre par de poços, injetor/produtor, utilizando a seguinte expressão matemática:

$$\lambda_{ij}^{sh} = \frac{\sqrt{MSE_{ij}}}{\sum_{j=1}^{N_p} |MSE_{ij}|}.$$

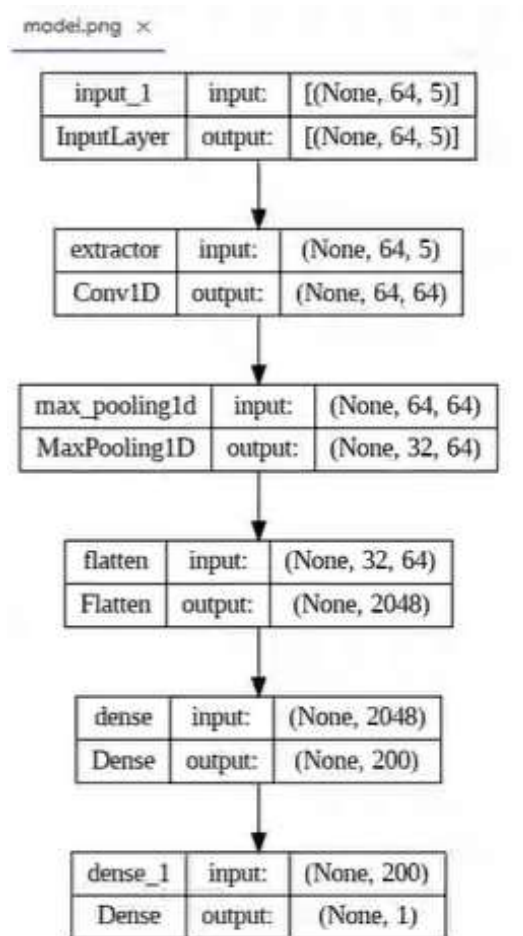
5.3 Gradient Importance – Método do Gradiente

Em uma imagem da permeabilidade do campo podemos observar uma área de maior interesse, baixa ou alta permeabilidade, e com estas informações é possível tomar uma decisão. Estas ideias podem ser generalizadas para o aprendizado automatizado e verificar a contribuição de cada entrada na saída final da previsão (Cerleani, 2020 e Cerleani, 2021) . A rede neural do gradiente Importance, é composto por uma célula e três portais: célula de memória, portal de entrada, portal de saída e portal do esquecimento. A célula de memória lembra dos valores dados através de intervalos arbitrários e os portais são como as camadas de entrada e saída da rede neural, fazendo com que a informação flua entre a rede.

Aqui a conectividade entre poços é dada pela importância do gradiente de cada entrada sobre a predição final e pode ser obtido pelos seguintes passos:

Informe valores iniciais dos pesos e viéses para conectar a arquitetura da rede neural convolutional (CNN). Especificamente escolhemos a CNN1D amplamente utilizada para processar dados unidimensional (e.g., series temporais). A CNN1D define camadas convolucionais para extrair informação local (pooling), e Flatten para reduzir a dimensionalidade dos dados e uma camada densa para os parâmetros de saída. Os parâmetros trabalhados no CNN1D podem ser vistos na figura 15 abaixo.

Figura 15 – Parâmetros utilizados no método do CNN1D



Fonte: A autora (2023)

Realizar o processo de treinamento para ajustar os parâmetros, pesos e viéses, da rede minimizando os dados observados e calculados da vazão líquida até atingir os parâmetros ótimos.

Extrair o gradiente ótimo após processo de treinamento para cada poço produtor com respeito ao injetor, $grad_importance_{ij}$, $i = 1 \dots N_i$ e $j = 1 \dots N_p$. Neste caso, o gradiente é obtido utilizando Diferenciação Automática no TensorFlow.

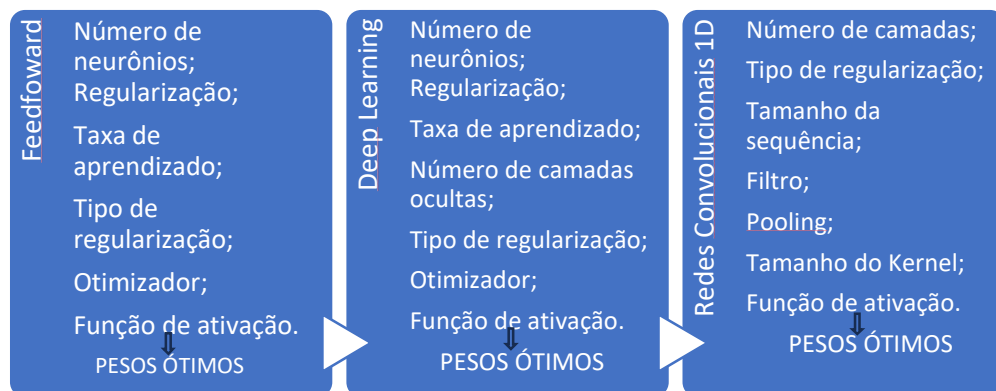
A conectividade entre o i -ésimo injetor e j -ésimo produtor é dado pela seguinte expressão matemática:

$$\lambda_{ij}^{grad} = \frac{grad_importance_{ij}}{\sum_{j=1}^{N_p} grad_importance_{ij}}.$$

5.4 Determinação dos Parâmetros para Cada Uma das Rede Neurais

Para cada uma das redes neurais citadas acima, foram utilizados parâmetros testes com um dos modelos de reservatório e fora comparado com os resultados obtidos pelo CRMP fazendo com que a rede de melhor aproximação com o valor real fosse escolhida e então aplicada ao outro modelo de reservatório. Na tabela 1, temos os parâmetros que foram modificados e escolhidos (pintados em cinza), bem como suas respectivas modificações em todas as estratégias. Também foi necessário informar os parâmetros para cada uma das opções presentes na figura 16 abaixo.

Figura 16 – parâmetros a serem informados pelos métodos a serem estudados



Fonte: A autora (2023)

Tabela 1 - Parâmetros de escolha da rede neural para o caso Feedforward

Regularização	Taxa de Aprendizado	Tipo de Função	Otimizador	Número de Neurônios
0,01	0,01	SIGMOID	RMSPROP	10
0,001	0,001	reLU	ADAM	50
0,0001	0,0001			100

Tabela 2 - parâmetros de escolha da rede neural para o caso do Embaralhamento

Número de Camadas	
Ocultas	Número de Neurônios 2ª Camada
2	10
3	50
	100

Tabela 3 - parâmetros de escolha da rede neural para o caso do Gradiente

Tamanho da Sequência	Pooling
24	pooling
48	maxpooling
96	

6 RESULTADOS E DISCUSSÕES

Neste estudo, dois *dataset* de modelos de reservatórios (*Synthetic two-phases model* (Cao et al., 2014) e *Brush Canyon Outcrop* (BCO by Oliveira, 2006) são utilizadas para validar as estratégias propostas. Os dados do histórico de produção dos poços produtores e injeção são obtidos executando o simulador de reservatórios comercial IMEX (CMG, 2021). Aqui, as vazões de injeção e produção líquida são medidas em condições de reservatório.

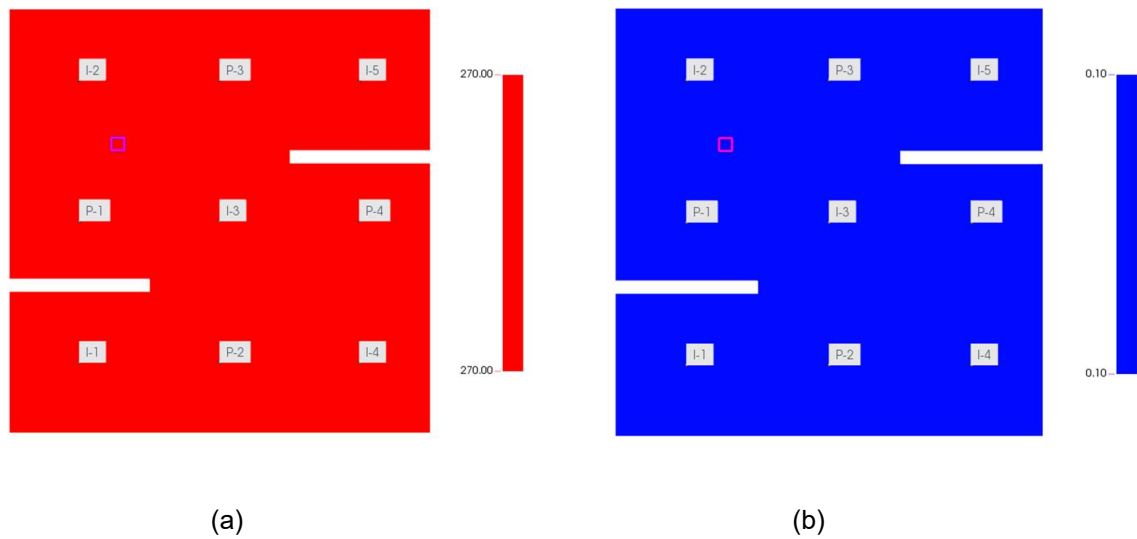
Nas estratégias de aprendizado automatizado para o treinamento, *Feedforward*, aprendizado profundo multicamadas e CNN1D, o critério de parada utilizado é dado pelo máximo número de épocas, 1000. A função de ativação estudada em todos os casos é a *rectified linear (ReLU)*.

Na estratégia CRMP, o algoritmo de otimização para resolver o problema acoplado é a programação quadrática sucessiva – SQP e sua derivada é aproximada usando o método das diferenças finitas. O erro máximo permitido na função objetivo é 10^{-4} .

6.1 Exemplo 1: Modelo Sintético de Duas Fases

O *dataset* utilizado está baseado em um modelo de reservatório bifásico simétrico que possui internamente duas falhas parciais. As células que modelam as falhas possuem porosidade, transmissibilidade e permeabilidade igual a zero (Fig. 17a). A permeabilidade horizontal do reservatório é constante, 270 millidarcy. A permeabilidade vertical é 10% da permeabilidade horizontal. A saturação inicial de água é 0.1 (Fig. 17b), a porosidade é 0.2. O modelo possui 5 camadas, com 33x33 células. A pressão média do reservatório é 500 psi. Uma descrição detalha do modelo pode ser encontrada em Cao et al., 2014.

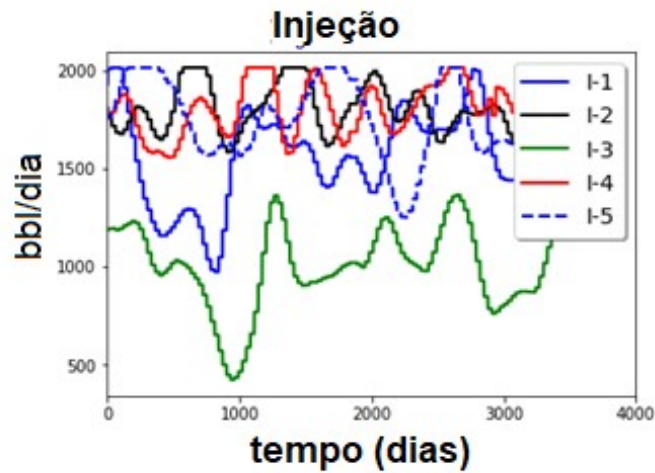
Figura 17 – Modelo sintético de duas fases: (a) campo de permeabilidade (b) saturação de água



Fonte: A autora (2023)

O reservatório possui 9 poços verticais, 4 produtores e 5 injetores. Os poços produtores são operados por bottom-hole pressure (BHP) e são mantidos fixos em 220 psi durante todo o período da simulação, 10 anos. No caso dos poços injetores, eles são operados por vazão. Os controles são gerados aleatoriamente com correlação temporal e tamanho de perturbação de 10% (Tueros et al., 2018) no intervalo $[0, 2000]$ bbl/dia e são alterados a cada 30 dias (Fig. 18). As respostas após simulação são recuperadas a cada um dia.

Figura 18 – Modelo sintético de duas fases: Taxa de entrada para todos os injetores mudando a cada 30 dias

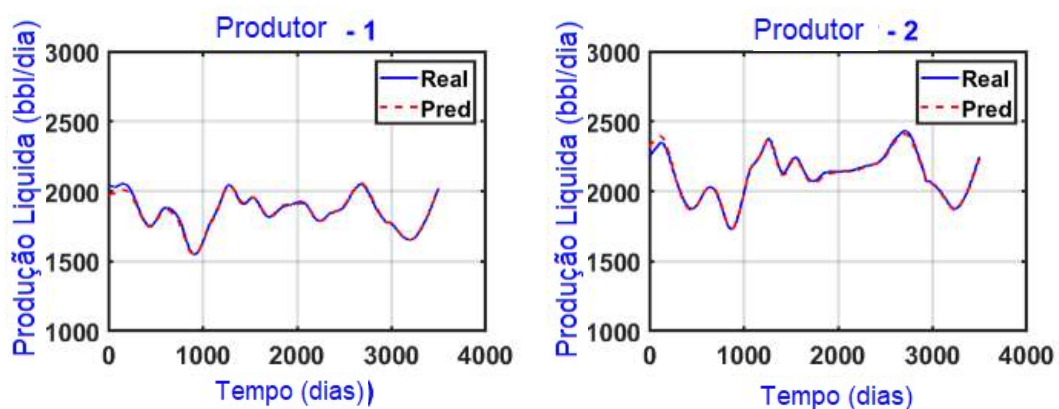


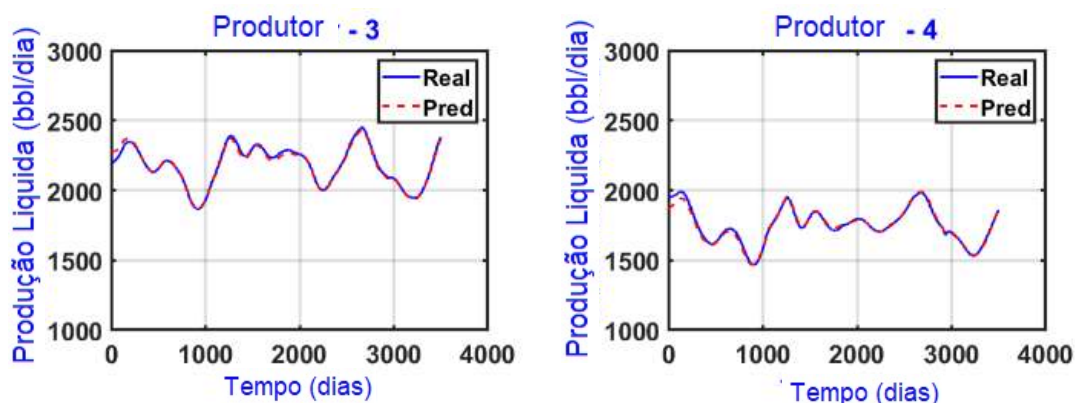
Fonte: A autora (2023)

6.1.1 Caso 1: Conectividade pela Modelo de Capacitância e Resistencia do Produtor - CRMP

O processo de otimização utilizou, 35 iterações e 1256 avaliações da função objetivo para encontrar o valor ótimo das conectividades. Na Fig. 19 exibimos o ajuste entre os dados observados e calculados da vazão líquida para os poços produtores obtidos pela estratégia CRMP.

Figura 19 – Modelo sintético de duas fases: predição da produção líquida e a real para todos os produtores





Fonte: A autora (2023)

Os valores das conectividades pela estratégia CRMP para cada par de poços, injetor/produtor, são exibidos na Tabela 4.

Tabela 4 – Modelo sintético de duas fases: Conectividades pelo CRMP

	P-1	P-2	P-3	P-4
I-1	0.2038	0.4328	0.1333	0.2300
I-2	0.3518	0.1650	0.3476	0.1357
I-3	0.2590	0.2569	0.2474	0.2367
I-4	0.1223	0.3718	0.1905	0.3154
I-5	0.2417	0.1287	0.4326	0.1970

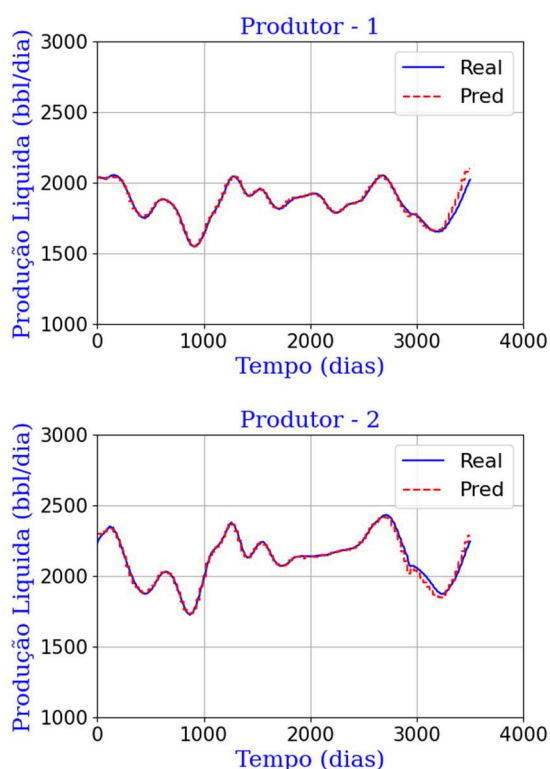
Na Fig. 17(a) temos observado que os poços I-1/P-1 e I-5/P-4 estão separados por uma falha e o valor das conectividades para ambos poços são 0.2038 e 0.1970, respectivamente. Os dois valores são próximos e concordantes devido a simetria do modelo de reservatório.

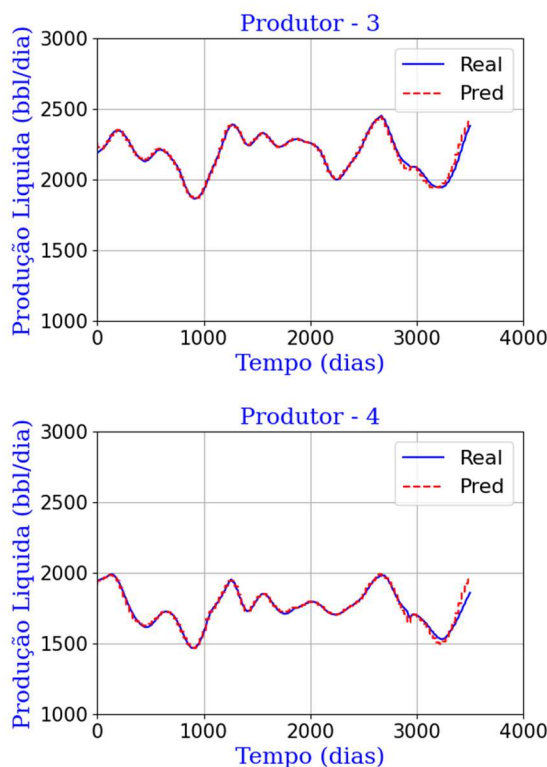
Em concordância com a Fig. 17(a), o poço injetor, I-1, tem uma influência direta sobre os poços produtores P-2, esse impacto é refletido no valor da conectividade entre ambos, 0.4328. Analogamente, a conectividade para os poços, I-5 e P-3, é 0.4326. Por outro lado, o injetor I-3 influi a todos os poços produtores, isto também é refletido no valor das conectividades, as quais são proporcionais. No caso de poços distantes, I-4 e P-1, o valor da conectividade é 0.1223. Em geral, o valor das conectividades pela estratégia CRMP serão utilizados para realizar comparações eficientes com os resultados das ANNs.

6.1.2 Caso 2: Conectividade pelo Feedforward

Na figura 20, mostramos o ajuste entre dados observados e calculados da vazão líquida pela estratégia *Feed Forward Backpropagation* (FFBackProp) para todos os poços produtores. Os hiper parâmetros no processo de treinamento são batch size de 64, 7 neurônios, regularização igual a 10^{-4} , learning rate igual a 10^{-4} . O número total de parâmetros treináveis para cada poço são 50. O erro obtido no processo de treinamento para cada um dos poços produtores, de P-1 a P-4, são 1.3%, 1.2%, 1.4% e 1.3%, respectivamente.

Figura 20 – Modelo sintético de duas fases: produção líquida real e prevista para todos os Produtores por meio do feedforward





Fonte: A autora (2023)

Os valores das conectividades entre poços são exibidos na Tabela 5. O valor da conectividade entre os poços, I-1e P-1, é 0.2550. No caso dos poços, I-5 e P-4, tem-se uma conectividade igual a 0.2105. Comparando ambos valores de conectividade eles são próximos devido a simetria do reservatório. Por outro lado, o valor da conectividade entre o injetor I-1 e produtor P-2 é 0.3618 e para o par, I-5 e P-3, é 0.3869. O valor das conectividades para pares de poços distantes, I-1/P-4 e I-5/P-1, são 0.2363 e 0.2105, respectivamente.

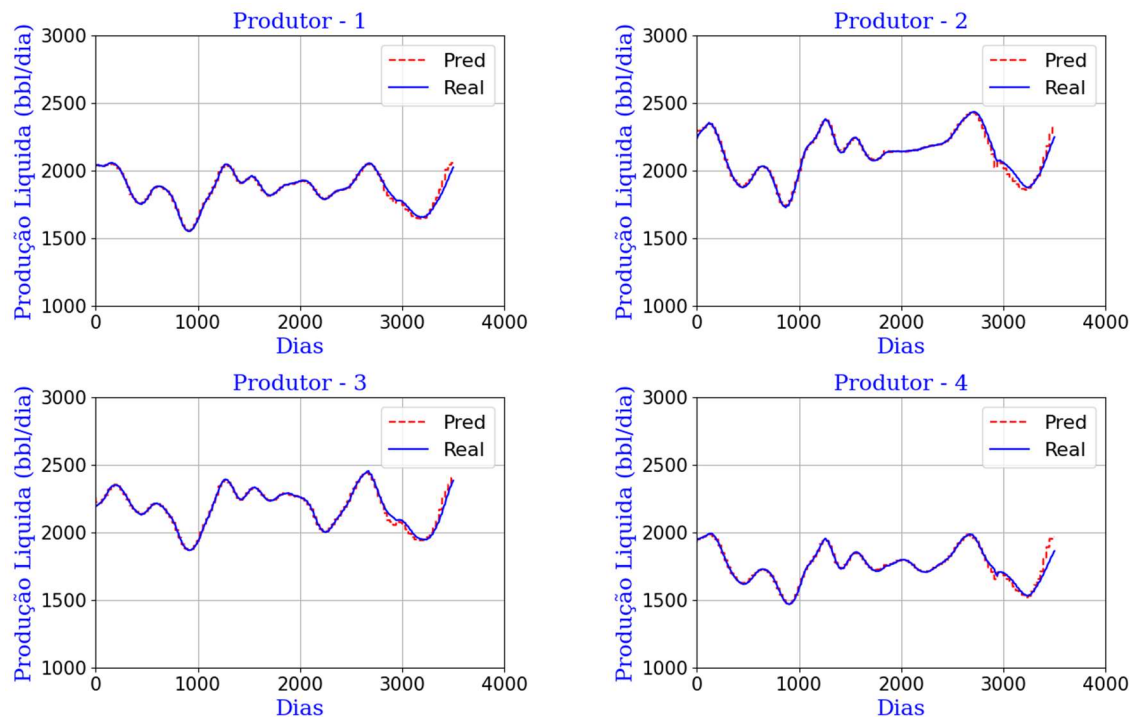
Tabela 5 – Modelo sintético de duas fases: Conectividades por feedfoward

	P-1	P-2	P-3	P-4
I-1	0.2550	0.3618	0.1467	0.2363
I-2	0.3483	0.1678	0.3251	0.1585
I-3	0.2902	0.1974	0.2212	0.2910
I-4	0.1995	0.2719	0.2234	0.3050
I-5	0.2685	0.1339	0.3869	0.2105

6.1.3 Caso 3: Conectividades por Embaralhamento – *Shuffling*

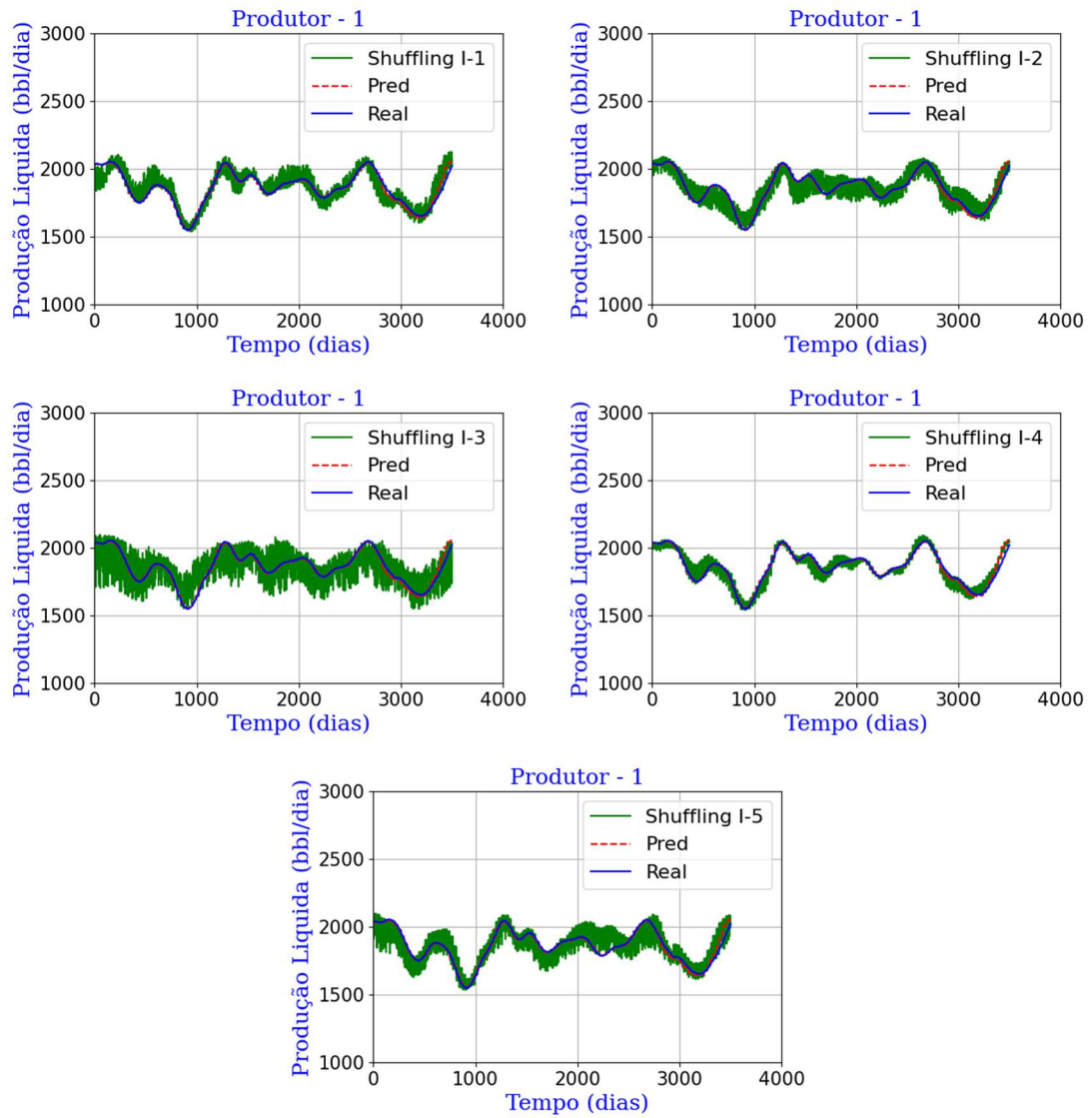
Na Fig. 22 exibimos resultado do ajuste entre da produção líquida observada e calculada utilizando e o impacto da estratégia do embaralhamento. Os hiper parâmetros utilizados no processo de treinamento são batch size de 64, duas camadas ocultas com 10 neurônios cada uma, *learning rate* de 10^{-4} . O número total de parâmetros treináveis para cada poço é 181. O erro obtido no ajustamento dos dados observados e calculados no processo de treinamento para todos os poços produtores de P-1 a P-4 são 5.7%, 4%, 4.6% e 6%, respectivamente.

Figura 21 - Modelo Sintético de Duas Fases: Produção líquida prevista e real para todos os Produtores pelo método de embaralhamento



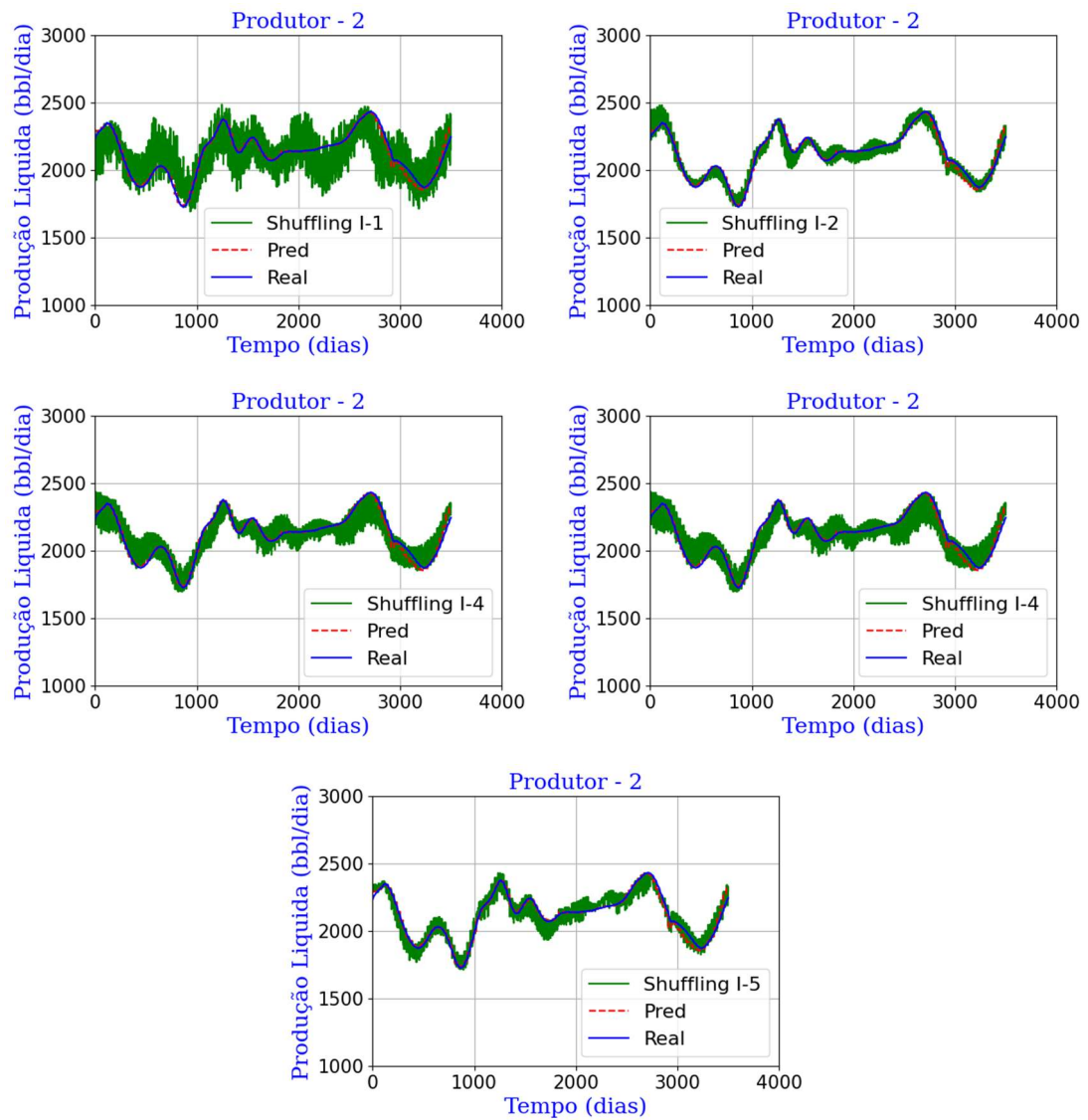
Fonte: A autora (2023)

Figura 22 – Modelo sintético de duas fases: a previsão da produção líquida e a real para o produtor 1



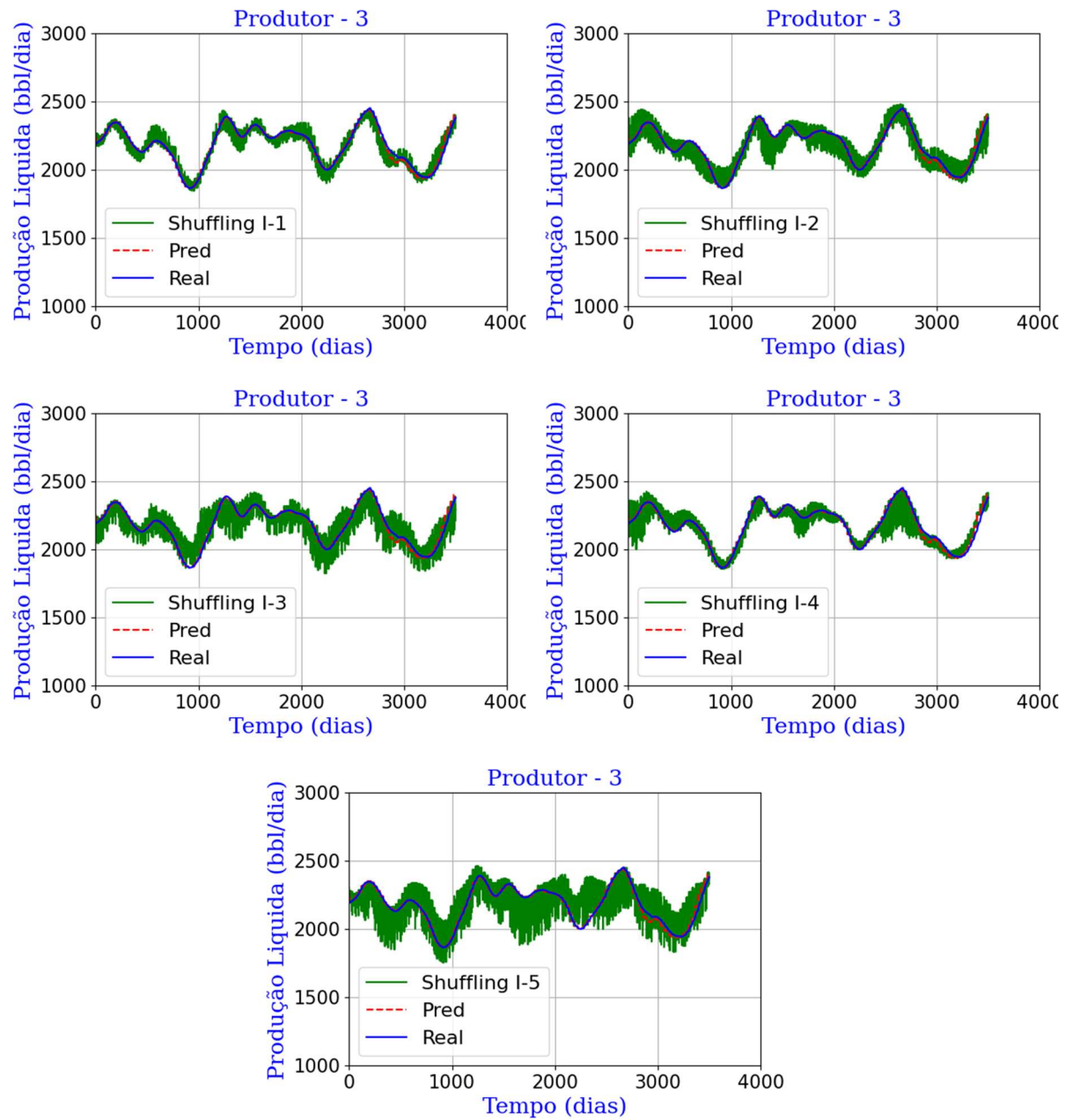
Fonte: A autora (2023)

Figura 23 – Modelo sintético de duas fases: a previsão da produção líquida e a real para o produtor 2



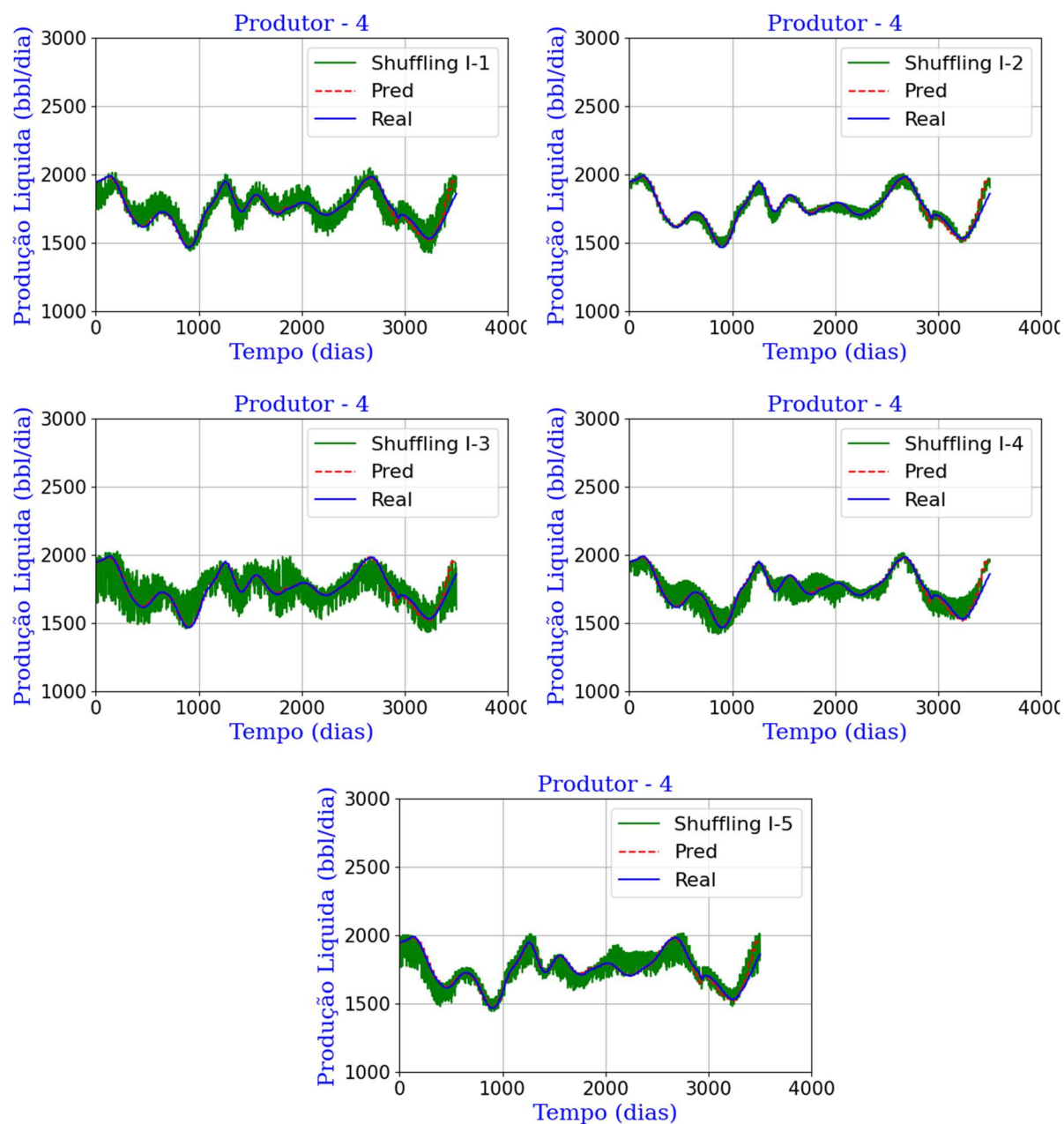
Fonte: A autora (2023)

Figura 24 - Modelo sintético de duas fases: a previsão da produção líquida e a real para o produtor 3



Fonte: A autora (2023)

Figura 25 - Modelo sintético de duas fases: a previsão da produção líquida e a real para o produtor 4



Fonte: A autora (2023)

Na Tabela 6 apresentamos os valores das conectividades para cada par de poços, injetor e produtor, pela estratégia do embaralhamento. O embaralhamento do poço injetor I-1 sobre a vazão líquida do produtor P-3, é refletido no valor da conectividade entre ambos, 0.1636. O embaralhamento dos controles do injetor I-2, gera um impacto proporcional sobre os produtores P-1 e P-3 e suas conectividades são 0.3433 e 0.3045, respectivamente. O embaralhamento das vazões do injetor I-3 impacta

proporcionalmente a todos os produtores, de P-1 a P-4, com valores de conectividade igual a 0.2916, 0.1466, 0.2498 e 0.3120, respectivamente. Enquanto que a conectividade entre o injetor I-1 e produtor P-4 é 0.2499.

Tabela 6 – Modelo Sintético de Duas Fases: Conectividade por *Shuffling*

	P-1	P-2	P-3	P-4
I-1	0.2363	0.3501	0.1636	0.2499
I-2	0.3433	0.1356	0.3045	0.2166
I-3	0.2916	0.1466	0.2498	0.3120
I-4	0.1859	0.2909	0.1931	0.3301
I-5	0.2197	0.1403	0.4184	0.2217

6.1.4 Caso 4: Conectividade pelo gradiente

A seguir exibimos os resultados obtidos pelo impacto do gradiente após técnica de treinamento. Neste caso, o processo de treinamento é obtido pela rede neural convolutional unidimensional, CNN1D. A arquitetura proposta utiliza 8 camadas, uma camada para os parâmetros de entrada, duas camadas para o Conv1D, uma camada de Maxpool1D, uma camada de *Flatten*, uma camada densa e uma camada para os parâmetros da saída. O filtro utilizado nas camadas convolucionais foi de 64, o tamanho do kernel é de 50. O número de parâmetros treináveis para cada poço são de 737.

Na Tabela 7 mostramos os valores das conectividades entre poços produtores e injetores pela estratégia do impacto dos gradientes utilizando a estratégia CNN1D para o treinamento dos dados. O valor da conectividade entre os poços, I-1e P-1, é 0.1797. No caso dos poços, I-5 e P-4, tem-se uma conectividade igual a 0.1933. Por outro lado, o valor da conectividade entre o injetor I-1 e produtor P-2 é 0.3390 e para o par, I-5 e P-3, é 0.4499. O valor das conectividades para poços distantes, I-1/P-4 e I-5/P-1, são 0.2290 e 0.2211, respectivamente.

Tabela 7 – Modelo Sintético de Duas Fases: Conectividade por CNN1D

	P-1	P-2	P-3	P-4
--	-----	-----	-----	-----

I-1	0.1797	0.4486	0.1424	0.2290
I-2	0.3997	0.1099	0.3363	0.1539
I-3	0.2744	0.2299	0.2455	0.2500
I-4	0.1896	0.3106	0.1462	0.3534
I-5	0.2211	0.1356	0.4499	0.1933

Na Tabela 8, comparamos o valor da conectividade do produtor P-1 com respeito aos injetores pelas diferentes estratégias propostas. O erro da conectividade entre o CRMP e as outras estratégias, *Feedfoward*, *Shuffling* e CNN1D, para o injetor I-1 são 25%, 16% e 11%, respectivamente. No caso da conectividade do injetor I-3, o erro de comparar o CRMP versus *Feedfoward*, *Shuffling* e CNN1D são de 12%, 12.5% e 6%, respectivamente. Analogamente, analisamos a conectividade do injetor I-5, obtendo os seguintes erros, 11%, 9% e 8.5%, respectivamente. Por fim, podemos concluir que a estratégia CNN1D mostra melhor aproximação no cálculo das conectividades.

Tabela 8 – Modelo Sintético de Duas Fases: Comparação de conectividades com o produtor 1 pelas diferentes técnicas

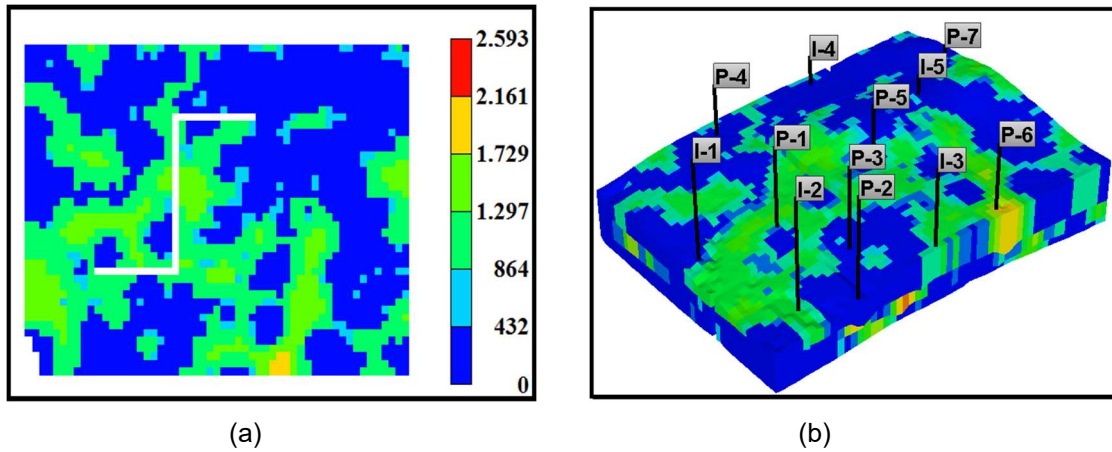
	CRMP	FFBackProp	Shuffling	CNN1D
I-1	0.2038	0.2550	0.2363	0.1797
I-2	0.3518	0.3483	0.3433	0.3997
I-3	0.2590	0.2902	0.2916	0.2744
I-4	0.1223	0.1995	0.1859	0.1896
I-5	0.2417	0.2685	0.2197	0.2211

6.2 Exemplo 2: Modelo Brush Canyon Outcrop - (BCO)

O segundo conjunto de dados usado é baseado no modelo BCO, que possui uma falha de vedação interna. As células que modelam a falha têm permeabilidade, porosidade e transmissibilidade definidas como zero. O modelo possui seis camadas, com 55x53 células. A permeabilidade horizontal varia de 157 a 2593 milidarcies (Figura 26a). A permeabilidade vertical é 30% da permeabilidade horizontal, a porosidade varia entre 16 e 28%, e a viscosidade é de 0,11 centipoise. O reservatório possui doze poços, sete produtores e cinco injetores (Figura 26b).

A vida útil do reservatório é de 3000 dias. Os poços produtores têm uma faixa operacional entre 4500 e 5000 Kpa, enquanto a taxa de injeção tem uma faixa operacional entre 0 e 2500 m³/dia. Os controles são alterados a cada seis meses, gerando assim 40 ciclos de controle.

Figura 26 – Modelo Brush Canyon Outcrop model: Permeabilidade (a) e localização de poços (b)

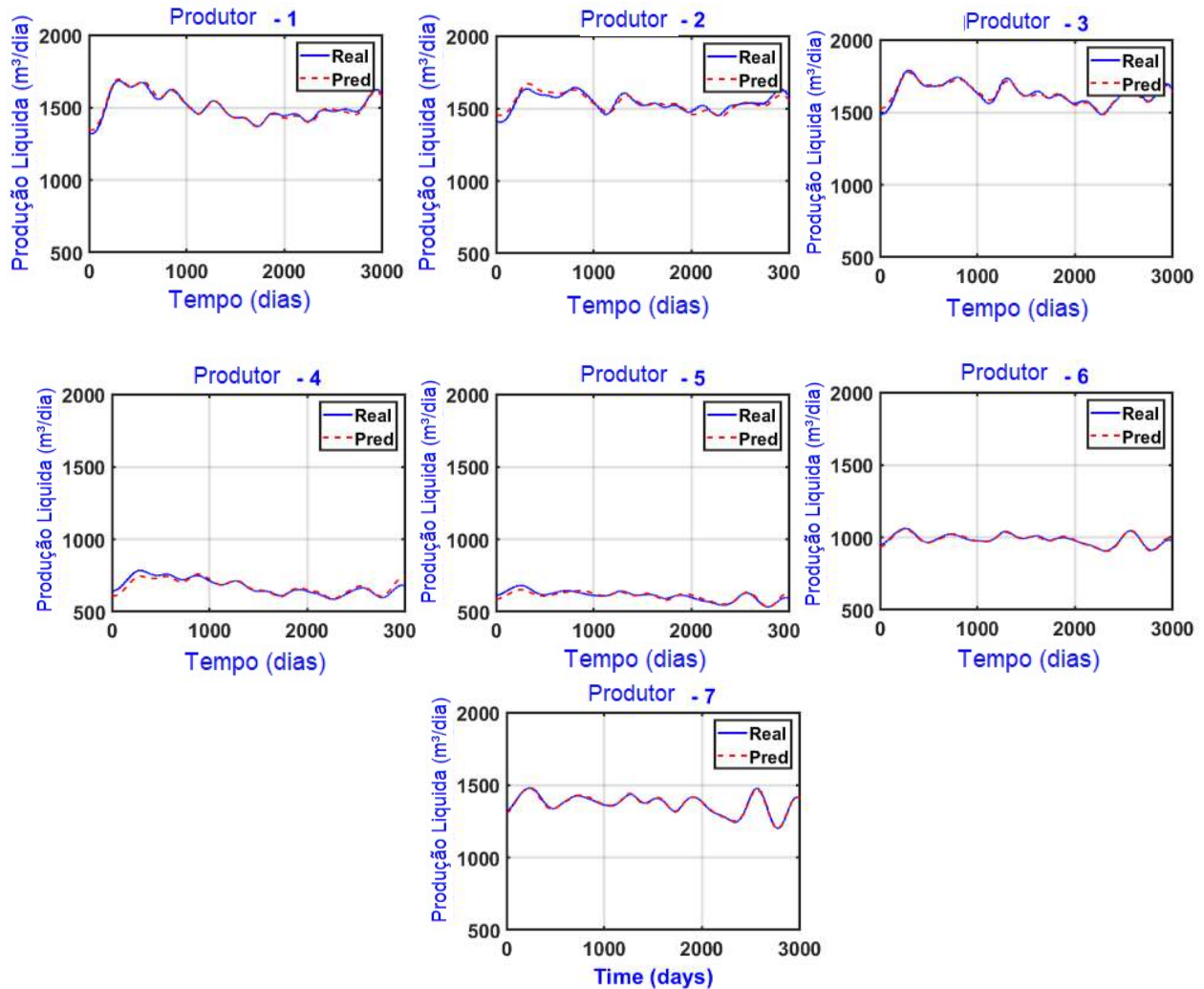


Fonte: A autora (2023)

6.2.1 Caso 1: Conectividade pela Modelo de Capacitância e Resistencia do Produtor - CRMP

Na Fig. 27 mostramos que o modelo CRMP captura a dinâmica do reservatório para a vazão líquida de todos os poços produtores. No processo do cálculo das conectividades foram utilizadas, 35 iterações e 1256 avaliações de função objetivo.

Figura 27 – Modelo BCO: Predição da vazão líquida e vazão real pelo CRMP



Fonte: A autora (2023)

Na tabela 9 exibimos o valor das conectividades para os poços injetores e produtores. No caso do injetor I-1 notamos que a maior conectividade é com respeito ao produtor P-1, 0.3919. A conectividade entre o injetor I-1e produtor P-5 ambos separados pela falha tem um valor igual a 0.0331. No caso do par I-2/P-2 o valor da

conectividade é de 0.2110 e para o poço distante e separado pela falha, I-2/P-4, tem conectividade igual a 0.0264. A conectividade entre o par I-3/P-1 é de 0.1198. O produtor P-1 também é influenciado pelos injetores I-4 e I-5, e suas conectividades são 0.1230 e 0.08, respectivamente.

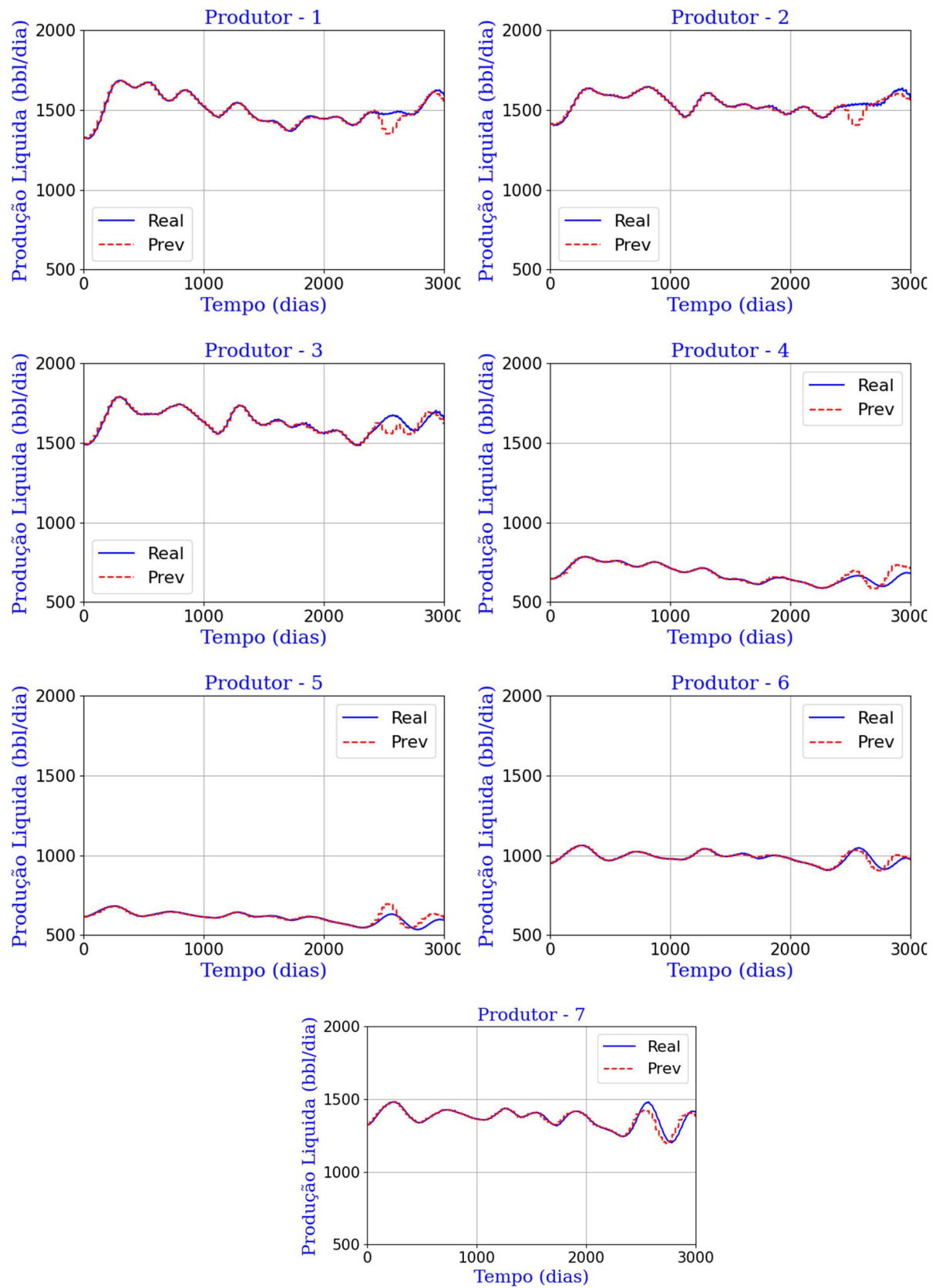
Tabela 9 – Modelo Brush Canyon Outcrop Conectividades pelo CRMP.

	P-1	P-2	P-3	P-4	P-5	P-6	P-7
I-1	0.3919	0.1701	0.1503	0.1115	0.0331	0.0598	0.0833
I-2	0.2110	0.3620	0.2537	0.0262	0.0268	0.0667	0.0536
I-3	0.1198	0.2039	0.2306	0.0562	0.0889	0.1687	0.1318
I-4	0.1230	0.0614	0.1159	0.1818	0.1193	0.1360	0.2626
I-5	0.0800	0.1270	0.2012	0.0342	0.0962	0.1556	0.3057

6.2.2 Caso 2: Conectividade pelo Feedforward

Na Fig. 28, exibimos o impacto da estratégia do feedforward no ajuste da vazão observada e calculada dos poços produtores. Os hiper parâmetros no processo de treinamento são *batch size* de 64, 50 neurônios, regularização igual a 10^{-4} , *learning rate* igual a 10^{-4} . O número total de parâmetros treináveis para cada poço são 351. O erro obtido no processo de treinamento para cada um dos poços produtores, de P-1 a P-7, varia de 1.2% a 1.6%.

Figura 28 – Modelo Brush Canyon Outcrop: Produção líquida prevista e real para todos os Produtores pelo método de retropropagação



Fonte: A Autora (2023)

Na tabela 10 mostramos o valor das conectividades para o modelo BCO pela estratégia *feedforward*. A seguir analisaremos como os poços injetores, de I-1 a I-5, impactam sobre o poço produtor P-1 a traves do valor das conectividades. O valor da conectividade entre o par I-1/P-1 é de 0.2217, o valor obtido é proporcional à distância entre ambos poços. No caso do injetor I-3 notamos que o valor da conectividade entre eles é de 0.0758 este valor é baixo devido à baixa permeabilidade entre eles. No caso do injetor I-4 e I-5 a conectividade com o produtor P-1 não é bem representada quando comparada com o valor obtido pela estratégia CRMP. O injetor I-3 afeta diretamente ao produtor P-6 e sua conectividade, 0.2445, é representativa em concordância com a distribuição de permeabilidade do reservatório. Por outro lado, a conectividade dos outros poços está diretamente relacionada com a locação dos poços.

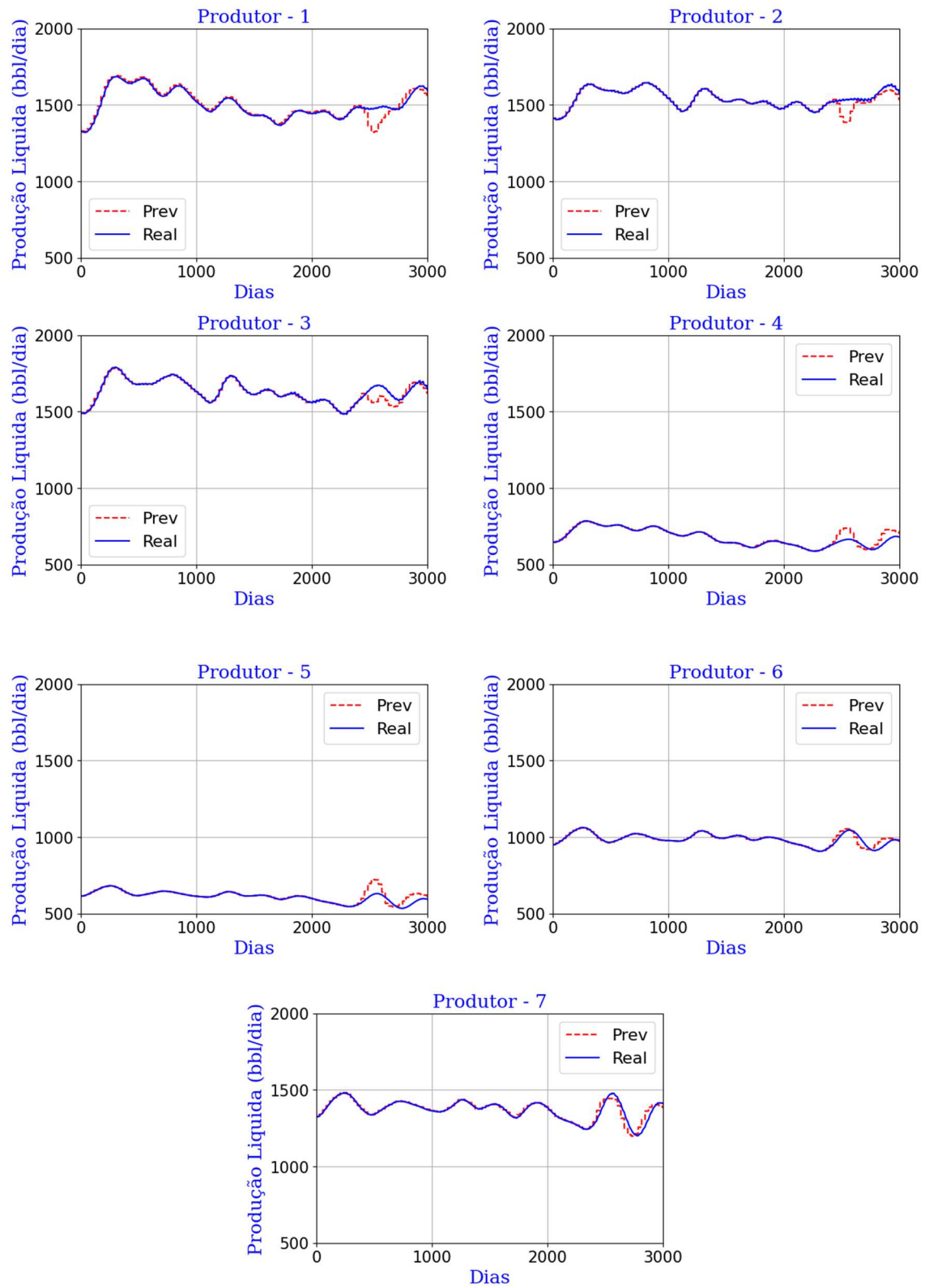
Tabela 10 – Modelo Brush Canyon Outcrop model: Conectividade pelo feedforward.

	P-1	P-2	P-3	P-4	P-5	P-6	P-7
I-1	0.2217	0.3263	0.2724	0.0536	0.0290	0.0477	0.0493
I-2	0.1268	0.2633	0.1933	0.0743	0.0501	0.1725	0.1198
I-3	0.0758	0.0577	0.2273	0.0380	0.2369	0.2445	0.1198
I-4	0.0036	0.1333	0.0298	0.2956	0.1769	0.1498	0.2110
I-5	0.0184	0.1066	0.0305	0.0786	0.2352	0.2507	0.2799

6.2.3 Caso 3: Conectividade pelo Embaralhamento - *Shuffling*

Na Figura 29 mostramos o ajuste entre a produção líquida observada e calculada obtido pela da estratégia do embaralhamento. Os hiper parâmetros utilizados no processo de treinamento são *batch size* de 64, duas camadas ocultas com 50 neurônios cada uma, *learning rate* 10^{-4} . O número total de parâmetros treináveis para cada poço é 2901. O erro obtido no ajustamento dos dados observados e calculados no processo de treinamento para todos os poços produtores de P-1 a P-7 varia entre 0.011 e 0.015.

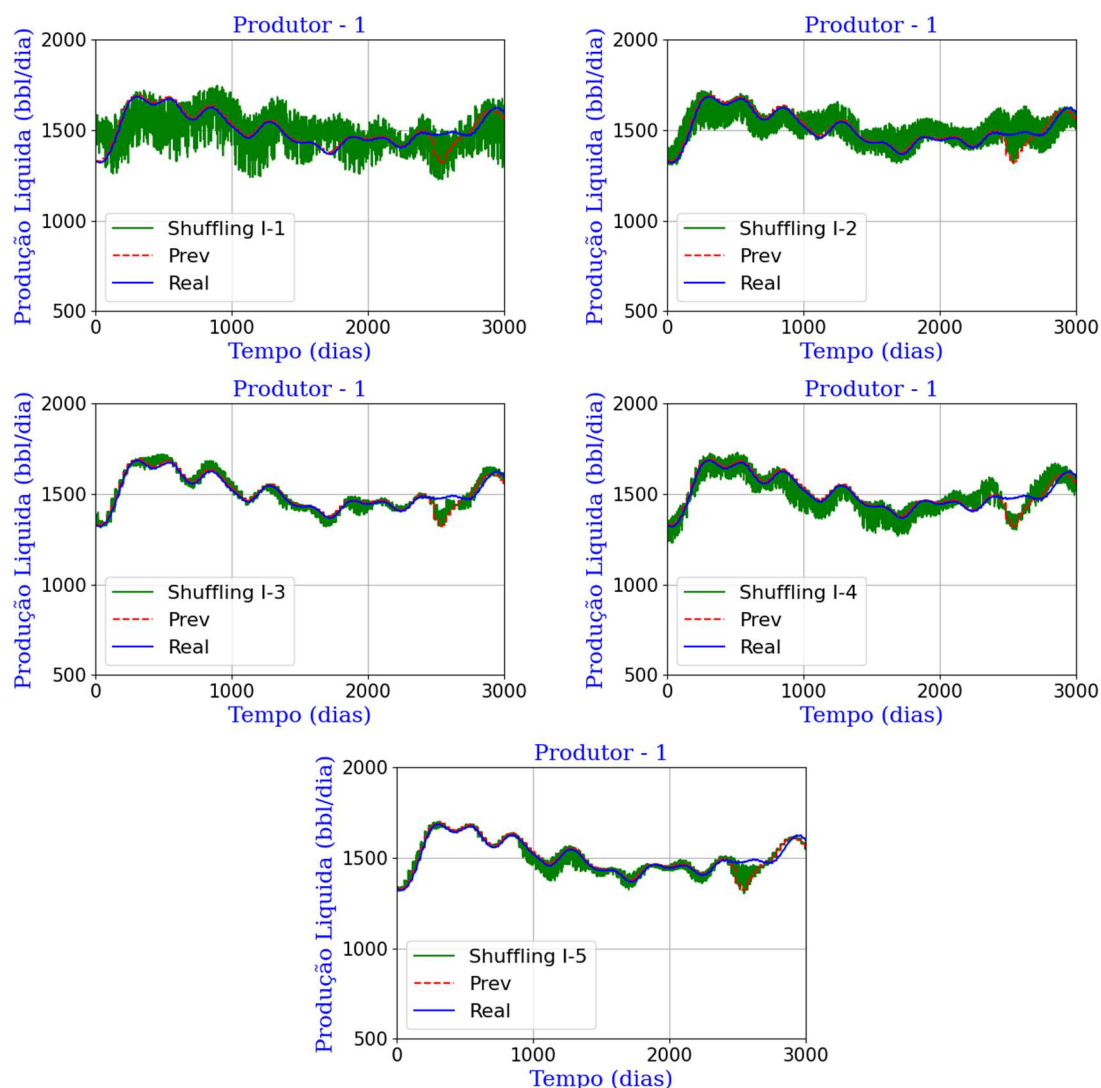
Figura 29 – Modelo Brush Canyon Outcrop: Produção líquida prevista e real para todos os Produtores pelo método de embaralhamento



Fonte: A autora (2023)

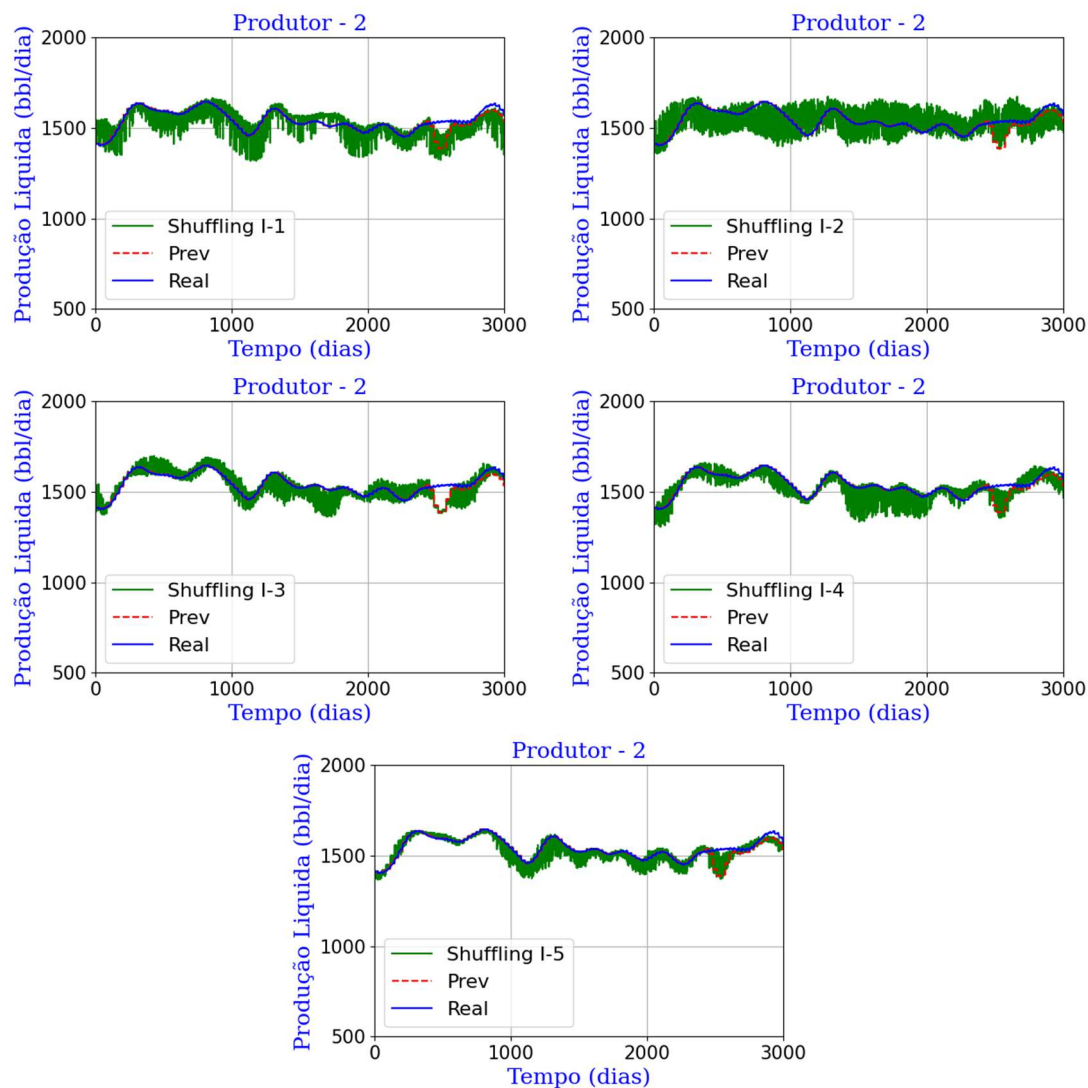
Na Figura 30 mostramos o impacto do embaralhamento dos controles dos injetores com respeito ao produtor P-1. Notemos que, o maior impacto está dado pelos injetores I-1 e I-2, enquanto que os outros injetores impactam em menor proporção, este impacto é concordante com a física do reservatório.

Figura 30 – Modelo Brush Canyon Outcrop: Produção líquida prevista e real para o produtor 1 e todos os injetores



Fonte: A autora (2023)

Figura 31 – Modelo Brush Canyon Outcrop: Produção líquida prevista e real para o produtor 2 e todos os injetores



Fonte: A autora (2023)

Na Tabela 11 apresentamos o valor das conectividades para o modelo BCO pela estratégia do embaralhamento dos controles nos poços injetores. Os poços P-1, P-2 e P-6 apresentam um alto valor de conectividade com os injetores I-1, I-2 e I-3. O injetor I-3 exibem baixo valor de conectividade com os produtores P-1 e P-4, devido as características do reservatório, distância e canais de baixa permeabilidade. Os injetores I-4 e I-5 apresentam baixo valor de conectividade com o produtor P-1. O valor das conectividades dos outros poços produtores e injetores está relacionada diretamente as propriedades físicas do reservatório.

Tabela 11 - Modelo Brush Canyon Outcrop: Conectividades pelo *Shuffling*

	P-1	P-2	P-3	P-4	P-5	P-6	P-7
I-1	0.2078	0.1629	0.1200	0.1280	0.1472	0.1218	0.1122
I-2	0.1119	0.2109	0.2046	0.1169	0.1225	0.1332	0.1000
I-3	0.0735	0.1581	0.1755	0.0909	0.1523	0.2176	0.1322
I-4	0.0862	0.1005	0.0984	0.2053	0.1856	0.1536	0.1704
I-5	0.0611	0.1203	0.1053	0.1079	0.1781	0.1933	0.2340

6.2.4 Caso 4: Conectividade pelo Gradient Importance – Gradiente

Na Tabela 12 exibimos o valor das conectividades para cada poço do modelo BCO pela estratégia da importância do gradiente. O maior valor de conectividade do produtor P-1 está associado como os injetores I-1 e I-2. O injetor I-2 impacta diretamente em 23% a produção líquida do produtor P-5 devido ao canal de permeabilidade entre eles. Por outro lado, os injetores de I-3 e I-4 possuem baixa conectividade com o produtor P-1.

Tabela 12 - Modelo Brush Canyon Outcrop: Conectividades pelo CNN1D.

	P-1	P-2	P-3	P-4	P-5	P-6	P-7
I-1	0.2163	0.1128	0.1449	0.1748	0.1058	0.1128	0.1326
I-2	0.1009	0.1449	0.1425	0.1232	0.2332	0.1329	0.1224
I-3	0.0750	0.0982	0.1262	0.1336	0.2179	0.1889	0.1603
I-4	0.0736	0.0984	0.1022	0.1439	0.2106	0.1469	0.2045
I-5	0.0809	0.1012	0.1202	0.1201	0.1508	0.1658	0.2611

Comparando as 4 estratégias, notamos que o valor das conectividades obtidos entre o produtor P-1 e os injetores são proporcionais. No caso de poços distantes, as estratégias de *Shuffling* e impacto do gradiente não representam bem o valor da conectividade.

Na Tabela 12, comparamos o valor da conectividade do produtor P-1 com respeito aos injetores pelas diferentes estratégias propostas. O erro da conectividade entre o CRMP e as outras estratégias, *Feedforward*, *Shuffling* e CNN1D, para o injetor I-1 são 25%, 16% e 11%, respectivamente. No caso da conectividade do injetor I-3, o erro de comparar o CRMP versus *Feedforward*, *Shuffling* e CNN1D são de 12%, 12.5%

e 6%, respectivamente. Analogamente, analisamos a conectividade do injetor I-5, obtendo os seguintes erros, 11%, 9% e 8.5%, respectivamente. Por fim, podemos concluir que a estratégia CNN1D mostra melhor aproximação no cálculo das conectividades.

7 CONCLUSÃO

Neste trabalho foram apresentadas duas metodologias, CRMP e ANNs, para calcular a conectividade entre par de poços produtores e injetores no reservatório de petróleo para um determinado tempo de produção. As metodologias propostas foram aplicadas a dois modelos de reservatório, modelo de duas fases e BCO.

Foi observado que o modelo de capacitância e resistência baseado nos produtores (CRMP) mostra boa qualidade da previsão da vazão líquida entre os dados observados e calculados. O algoritmo utilizado para resolver o problema acoplado e determinar as conectividades entre par de poços é a programação quadrática sucessiva (SQP).

Notamos que os valores das conectividades pela estratégia CRMP são proporcionais a geometria do reservatório, poços distantes e separados pela falha possuem baixa conectividade, enquanto que os outros poços possuem um valor de conectividade maior.

Foram ensaiados três tipos de arquitetura, *Feedforward*, Aprendizado profundo e Redes Neurais Convolucionais – CNN, para realizar o processo de treinamento minimizando a função de perda com objetivo de ajustar os pesos entre os neurônios de cada camada.

Da posse dos pesos ótimos foram propostas três estratégias, média ponderadas, erro médio e importância do gradiente, para inferir o valor das conectividades entre pares de poços injetores e produtores. A principal aplicação da posse das conectividades é o refinamento do método baseado em ensembles.

O modelo *feedforward* usou apenas uma camada oculta, resultando em resultados bem próximos ao nosso modelo de referência, o CRMP. Já o modelo de aprendizado profundo, necessitou apenas de duas camadas ocultas para que os resultados das conectividades se encontrem-se próximos aos do modelo CRMP. No caso das redes convolucionais, usamos o *maxpooling* como modelo de *pooling* para o funcionamento da rede neural, este, foi o mais acurado em relação aos valores de conectividades. Os notebooks utilizados por estes métodos encontram-se em anexo.

Finalmente, resultados das conectividades pelo impacto do gradiente mostram ser mais acurados e próximos aos resultados pela estratégia CRMP.

Sugestões de trabalhos futuros:

O cálculo das conectividades entre poços produtores e injetores tem sido essencial para monitoramento dos poços em reservatório, aumentar a produção de petróleo, reduzir a produção de água e evitar possíveis reativações de falhas, respectivamente. Uma vez em posse dos valores das conectividades, é possível combinar com outras estratégias:

Utilização dos valores das conectividades calculados pelos métodos e utilizar no refinamento com o método baseado em ensembles;

Cálculo das conectividades pelos métodos utilizando dados de campo de injeção e produção dos poços do reservatório e comparação dos valores reais de conectividade com os calculados pelos métodos estudados.

Detectar falhas de vedação ou regiões de baixa permeabilidade entre poços de injeção e produção usando apenas dados de produção, sem a necessidade de um modelo geológico completo.

Na otimização da produção, quando derivadas não estão disponíveis, a alternativa é usar o método baseado em conjunto. O método introduz correlações espúrias ao usar tamanhos pequenos de conjunto de controles. A conectividade pode ser usada para eliminar correlações entre poços distantes ou poços separados por falhas de vedação, refinando assim o cálculo do gradiente.

A partir dos valores de conectividade e do fluxo de injeção, é possível prever a produção de líquidos nos poços produtores e, em conjunto com modelos de fluxo fracionário, pode-se determinar o fluxo de óleo e água produzida.

REFERÊNCIAS

- Albertoni, A., Lake, L., 2003. Inferring interwell connectivity only from well-rate fluctuations in waterfloods. *SPE Reserv. Eval. Eng* 6(1), 6-16.
- ABBASSI, F. et al. Poromechanics of Fractured/Faulted Reservoirs During Fluid Injection Based on Continuum Damage Modeling and Machine Learning. *Natural Resources Research*, v. 32, n. 1, p. 413–430, 30 out. 2022.
- Artun, E. (2016). Characterization Reservoir Connectivity and Forecasting Waterflood Performance Using Data-Driven and Reduced-Physics Models. Paper presented at the SPE Western Regional Meeting, Anchorage, Alaska, USA, doi: <https://doi.org/10.2118/180488-MS>
- BRUCE, W. A. An Electrical Device for Analyzing Oil-reservoir Behavior. *Petroleum Technology*, v. 151, p. 112–124, 1943.
- Canning, A., Moulière-Reiser, D., Weiss, Y., Malkin, A., Phillip, E., Grinberg, N., Yehezkel, V. (2017, October 23). Neural networks approach to spectral enhancement. Society of Exploration Geophysicists.
- Cao F., Luo H., Lake L.W. (2015) Oil-rate forecast by inferring fractional-flow models from field data with Koval method combined with the capacitance/resistance model, *SPE Reserv. Eval. Eng.* 18, 4, 534–553.
- Cerleani M (2020) Feature importance with time series and recurrent neural network. In: *Towards data Science*.
- Cerleani M (2021) Advanced permutation importance to explain predictions. In: *Towards data Science*.
- Cheng, H., Vyatkin, V. Osipov, E. Zeng P. and Yu, H., "LSTM Based EFAST Global Sensitivity Analysis for Interwell Connectivity Evaluation Using Injection and Production Fluctuation Data," in *IEEE Access*, vol. 8, pp. 67289-67299, 2020, doi: 10.1109/ACCESS.2020.2985230.
- Computer Modeling Group LTD.: IMEX: User's Guide. Calgary – Canada (2020) Davudov, Davud, Malkov, Anton, and Ashwin Venkatraman. "Integration of Capacitance Resistance Model with Reservoir Simulation." Paper presented at the SPE Improved Oil Recovery Conference, Virtual, August 2020. doi: <https://doi.org/10.2118/200332-MS>
- Elzenary, M., Elkatatny, S., Abdelgawad, K. Z., Abdulraheem, A., Mahmoud, M., & Al-Shehri, D. (2018, August 16). New Technology to Evaluate Equivalent Circulating Density While Drilling Using Artificial Intelligence. Society of Petroleum Engineers. doi:10.2118/192282-MS

Ertekin, T.; Sun, Q. Artificial Intelligence Applications in Reservoir Engineering: A Status Check. *Energies* 2019, 12, 2897. <https://doi.org/10.3390/en12152897>

Holanda R.W., Gildin E., Jensen J.L. (2015) Improved waterflood analysis using the capacitance-resistance model within a control systems framework, Society of Petroleum Engineers. <https://doi.org/10.2118/177106-MS>

GAO, K. et al. Delineating faults at the soda lake geothermal field using machine learning. In: Proceedings of the 46th Workshop on Geothermal Reservoir Engineering, Stanford University, Stanford, CA, USA. 2021. p. 16-18.

GOMES, I.F. et al. Coupled Hydro-Mechanical Modeling of Fault Reactivation Problem in Petroleum Reservoir Considering the Fault Divided into Core and Damage Zones. In: AAPG/SEG International Conference & Exhibition, Cancun, Mexico, 2016.

GUIMARÃES L. J. N et al. Numerical modelling of CO₂ flow through deformable aquifers with the possibility of fault reactivation. In: Sixth International Congress on Environmental Geotechnics. New Delhi, India, 2010.

HAMZEH, A. Machine Learning Methods in Fault Detection. Dissertation (Master's Degree in Petroleum and Mining Engineering)- Politecnico di Torino, Torino, Italy, 2021.

Kaviani D., Valkó PP (2010) Inferring interwell connectivity using multiwell productivity index (MPI), J Pet Sci Eng. 73, 1, 48–58. <https://doi.org/10.1016/j.petro.2010.05.006>.

Kim, Tae Hyung. "Improvement of Reservoir Management Efficiency Using Stochastic Capacitance Resistance Model." Paper presented at the SPE Western Regional Meeting, San Jose, California, USA, April 2019. doi: <https://doi.org/10.2118/195322-MS>.

Kim, Y.D., Durlofsky, L.J. (2021) "A recurrent neural network-based proxy model for well-control optimization with nonlinear output constraints", SPE J., Vol. 26, N. 04, pp. 1837-1857.

LESUEUR, M.; POULET, T.; VEVEAKIS, M. Fault reactivation during fluid production, modelled as a multi-physics multi-scale instability. E3S Web of Conferences, v. 205, p. 03002, 2020.

Li, H., He, J., & Misra, S. (2018, September 17). Data-Driven In-Situ Geomechanical Characterization in Shale Reservoirs. Society of Petroleum Engineers. doi:10.2118/191400-MS

Liang, X., 2010. A simple model to infer interwell connectivity only from well-rate fluctuations in waterfloods. Journal of Petroleum Science and Engineering - J PET SCI ENGINEERING. 70. 35-43. 10.1016/j.petrol.2009.08.016.

Lima, Anny Virginia de Sousa. 2023. Aplicações de Técnicas de Machine Learning e Análise de Sensibilidade de Pressões de Injeção em Cenários de Reativação de

Zonas de Falhas Geológicas. Dissertação de Mestrado, Universidade Federal de Pernambuco.

Lins, H.K, Horowitz, B., Tueros, J.A.R.: Numerical Experience Using Capacitance Resistance Multilayered Models. CILAMCE 2017, Ibero-latin American Congress in Computational Methods in Engineering. Florianópolis, Brazil (2017). <http://dx.doi.org/10.20906/CPS/CILAMCE2017-0282> (in Portuguese).

LINS, H. K. D. F. Métodos Baseados em Modelos Dinâmicos de Capacitância e Resistência em Reservatórios de Petróleo de Múltiplas Camadas. Dissertação (Mestrado em Engenharia Civil). Universidade Federal de Pernambuco. Recife. 2018.

W. Liu, W.D. Liu, J. Gu, A machine learning method to infer inter-well connectivity using bottom-hole pressure data J. Energy Resour Technol., Technol., 142 (10) (2020). <https://doi.org/10.1115/1.4047304>

Mamghaderi, A. and Pourafshary, P., 2013. Water flooding performance prediction in layered reservoirs using improved capacitance-resistive model. J Petrol Sci Eng 108, 107–117 (2013). <http://doi.org/10.1016/j.petrol.2020.108151>

Mamghaderi, A., Aminshahidy B., Bazargan H., 2021. Prediction of waterflood performance using a modified capacitance-resistance model: A proxy with a time-correlated model error. J Petrol Sci Eng 198, 108152 (2021). <http://doi.org/10.1016/j.petrol.2013.06.006>

Moreno G.A. (2013) Multilayer capacitance–resistance model with dynamic connectivities, J. Pet. Sci. Eng. 109, 298–307. <https://doi.org/10.1016/j.petrol.2013.08.009>

Naudomsup, Nuntha, and Larry W. Lake. "Extension of Capacitance-Resistance Model to Tracer Flow for Determining Reservoir Properties." Paper presented at the SPE Annual Technical Conference and Exhibition, San Antonio, Texas, USA, October 2017. doi: <https://doi.org/10.2118/187410-MS>.

Nielsen, Michael A., "Neural Networks and Deep Learning", Determination Press, 2015. Acesso em 2023. < <http://neuralnetworksanddeeplearning.com> >

Olden, J. and Jackson, D. 2002. Illuminating the black box: a randomization approach for understanding variable contributions in artificial neural networks. Ecol Model, 154(1–2):135–150. doi: 10.1016/S0304-3800(02)00064-9.

Oliveira, D.F.B., 2006. Production Optimization Techniques for Petroleum Reservoirs: Derivate Free Approaches to Dynamic Rate Allocation for Injection and Production. Master thesis. Civil Engineering Department, UFPE, Recife, Brazil (in Portuguese).

Oliveira S. D., Horowitz, B., Tueros, J. A. R. (2021) Ensemble-based method with combined fractional flow model for waterflooding optimization. Oil Gas Sci. Technol. – Rev. IFP Energies Nouvelles, 76 (2021) 7 <https://doi.org/10.2516/ogst/2020090>

Panda M., Chopra A. (1998) An integrated approach to estimate well interactions, in: SPE India oil and gas conference and exhibition, Society of Petroleum Engineers. <https://doi.org/10.2118/39563-MS>.

PEREIRA, L. C. et al. Coupled hydro-mechanical fault reactivation analysis incorporating evidence theory for uncertainty quantification. *Computers and Geotechnics*, v. 56, p. 202–215, mar. 2014

Pereira, L. C. Quantificação de Incertezas Aplicada à Geomecânica de Reservatórios. 2015. 195f. Tese (Doutorado em Engenharia Civil) - Universidade Federal de Pernambuco, Recife, 2015.

ROSA, A.; CARVALHO, R. D. S.; XAVIER, J. A. D. Engenharia de reservatórios de petróleo. Rio de Janeiro: Interciência, 2011. p. 16; 90 - 120.

Ross, C. (2017, October 23). Improving resolution and clarity with neural networks. Society of Exploration Geophysicists.

RUTQVIST, J. The Geomechanics of CO₂ Storage in Deep Sedimentary Formations. *Geotechnical and Geological Engineering*, v. 30, n. 3, p. 525–551, 12 jan. 2012

Sayarpour, M., Zuluaga, E., Kabir, C.S., Lake, L.W.: The use of capacitance-resistive models for rapid estimation of waterflood performance and optimization. *J Petrol Sci Eng* 69 (3–4), 227–238 (2009). <http://doi.org/10.1016/j.petrol.2009.09.006>

Tueros, J.A.R., Horowitz, B., Willmersdorf, R., Oliveira, D.: Non-distance-based localization techniques for ensemble-based waterflooding optimization. *J Petrol Sci Eng*, 170, 440-452 (2018). <http://doi.org/10.1016/j.petrol.2018.06.089>

Weber, D.: The Use of Capacitance-resistance Models to Optimize Injection Allocation and Well Location in Water Floods. The University of Texas at Austin Ph.D. dissertation (2009)

Yousef A.A., Gentil P.H., Jensen J.L., Lake L.W. (2005) A capacitance model to infer interwell connectivity from production and injection rate fluctuations, in: SPE Annual Technical Conference and Exhibition, Society of Petroleum Engineers. <https://doi.org/10.2118/95322-MS>.

Yu, J. and Jahandideh, A. and Jafarpour, B. (2020). Engineering Design of Neural Network Architectures for Estimation of Inter-Well Connectivity and Production Performance. *European Association of Geoscientists & Engineers*. ISSN 2214-4609. <https://doi.org/10.3997/2214-4609.202035191>.

YUAN, Z. et al. An enhanced fault-detection method based on adaptive spectral decomposition and super-resolution deep learning. *Interpretation*, v. 7, n. 3, p. T713–T725, 1 ago. 2019.

ZHU, D. et al. Fault Detection by Using Instance Segmentation. In: International Petroleum Technology Conference. OnePetro, 2021.

ANEXO A – ANEXO DO NOTEBOOK DO MÉTODO *FEEDFORWARD*

```
''' Neural network with keras tutorial - Artun strategy'''

import pandas as pd
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt

from sklearn import preprocessing
from tensorflow.keras import regularizers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.callbacks import EarlyStopping

#''' 0 imprime e 1 = não imprime os dados por etapas para serem
verificados'''

input_prior= 1

#''' numero de injetores'''

ninj=5

#''' numero de produtores'''

nprod=7

#''' número de epocas no processo'''

epochs_= 500

#''' número de bach no processo'''

b_size =64

#''' tamanho de regularization'''

reg=0.0001

# "Learning rate "

lr=0.001
```

```

#''' numero de neuronios'''

neurons=50

#''' Well inputs and output (at analyze) '''

columns = ['I-1','I-2','I-3','I-4','I-5']

wells=[['P-1'],['P-2'],['P-3'],['P-4'],['P-5'],['P-6'],['P-7']]

# Matriz nula de conectividades

Matrixconectividade=np.zeros((ninj,nprod))

#''' 1.- load the dataset '''

#dataset=pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/RafaelDataSet3.dat',delimiter='\s+')
dataset=pd.read_csv('/content/BCODataSet.dat',delimiter='\s+')

...

#''' 2.- define the keras model '''

model=Sequential()
model.add(Dense(neurons, input_dim=ninj, activation='relu',
kernel_regularizer=regularizers.l2(reg)))
model.add(Dense(1,kernel_regularizer=regularizers.l2(reg)))

...

```

ANEXO B – ANEXO DO NOTEBOOK DO MÉTODO DO *SHUFFLING* OU EMBARALHAMENTO

```

''' Neural network with keras tutorial '''

import pandas as pd
import numpy as np
import tensorflow as tf
import matplotlib.pyplot as plt
from sklearn import metrics
from sklearn import preprocessing
from sklearn.utils import shuffle
from tensorflow.keras import regularizers
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.metrics import mean_squared_error
from math import sqrt
from random import seed

# seed random number generator
seed(1)

#'''0 não imprime e 1 = imprime os dados por etapas para serem
verificados, '''

input_prior= 1

#''' numero de injetores'''

ninj=5

#''' numero de produtores'''

nprod=7

```

```

''' numero de neuronios'''

neurons=50

''' número de epocas no processo'''
epochs_= 500

''' número de bach no processo'''
b_size =64

''' numero de paciencia na parada'''

paciencia=100

''' tamanho de regularization'''

reg=0.001 # bh =0.01

# Learning rate
lr=0.001

''' Well inputs and output (at analize) '''

columns = ['I-1','I-2','I-3','I-4','I-5']

wells=[['P-1'],['P-2'],['P-3'],['P-4'],['P-5'],['P-6'],['P-7']]

# Matriz nula de conectividades

Matrixconectividade=np.zeros((ninj,nprod))

''' 1.- Load the dataset '''

```

```
#dataset=pd.read_csv('RAFAELDataSet2.dat',delimiter='\s+')
dataset=pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/BCODataset.dat',delimiter='\s+')

```

```
pd.set_option('display.max_columns', 6)
pd.set_option('display.max_rows', 10)

```

```
idx=1

```

```
cont=0

```

```
# Initial

```

```
for w in wells:
    input_dataset=dataset[columns].copy()
    output_dataset=dataset[w].copy()

```

```
'''call functions '''

```

```
scalerM=preprocessing.MinMaxScaler()
scalerS=preprocessing.MinMaxScaler()

```

```
''' Normalizando a saída '''

```

```
tlabels=scalerM.fit_transform(output_dataset)

```

```
''' Normalizando as variáveis de entrada '''

```

```
tdata=scalerS.fit_transform(input_dataset)

```

```

''' 2.- define the keras model '''
model=Sequential()
model.add(Dense(neurons, input_dim=ninj, activation='relu',
kernel_regularizer=regularizers.l2(reg)))
model.add(Dense(neurons, activation='relu'))
#model.add(Dense(10, activation='relu'))
model.add(Dense(1,kernel_regularizer=regularizers.l2(reg)))

''' 3.- compile the keras model '''
#optimizer=tf.keras.optimizers.RMSprop(lr)
optimizer=tf.keras.optimizers.Adam(lr)
model.compile(loss='mse', optimizer=optimizer, metrics=['mae','mse'])
if input_prior == 1:
    print(model.summary())

''' Maxima paciencia no erro'''

Estop=tf.keras.callbacks.EarlyStopping(monitor='loss',patience=paciencia)
history=model.fit(tdata,tlabels,epochs=epochs_, batch_size=b_size,
validation_split=0.2,verbose=1)

'''plot evolution '''
font = {'family': 'serif','color':'darkblue','weight':'normal','size':
14,}
hist=pd.DataFrame(history.history)
hist['epoch']=history.epoch
plt.figure()
#plt.title('Epoch',fontdict=font)
plt.xlabel('Epoch',fontdict=font)
plt.ylabel('Mean abs Error',fontdict=font)
plt.plot(hist['epoch'],hist['mae'],label='Train Error')
plt.plot(hist['epoch'],hist['val_mae'],label='Val Error')
plt.legend()

```

```

plt.show()

# '''plot fit '''
train_predict0=model.predict(tdata).flatten()
train_predict1=train_predict0.reshape(train_predict0.shape[0],1)

train_predict = scalerM.inverse_transform(train_predict1)

#bestsol=np.zeros((nprod,train_predict.shape[0]))

#bestsol[cont,:]=np.transpose(train_predict)
#cont=cont+1

font = {'family': 'serif','color':'blue','weight':'normal','size': 18,}

plt.figure(figsize=(6,4))

plt.plot(train_predict,linestyle='dashed',c='r',label='Pred',linewidth=1.5
, markersize=1)
plt.title('Producer - '+str(idx), fontdict=font)
plt.plot(output_dataset,c='b',label='Real',linewidth=1.5, markersize=1)
plt.xlabel('Days',fontdict=font)
plt.ylabel('Liquid Rate (bbl/day)',fontdict=font)
plt.rcParams['xtick.labelsize'] = 20
plt.rcParams['ytick.labelsize'] = 20
plt.ylim([500, 2000])
plt.xlim([0, 3000])
plt.xticks(fontsize=15)
plt.yticks(fontsize=15)
plt.legend(fontsize=16)
plt.grid()

```



```

plt.show()
score =
np.sqrt(metrics.mean_squared_error(train_predict,output_dataset))

# ''' 4.- Connectivity by Root Mean Square d Error'''

# ''' Criando perturbação em todas as colunas '''
yshuffle=shuffle(input_dataset)

#rms=np.zeros((ninj,1))

for i in range(ninj):

    input_dataset=dataset[columns].copy()
    input_dataset['I-'+str(i+1)]=yshuffle['I-'+str(i+1)].values

# ''' Normalizando as variaveis de entrada '''

tdata_shuff=scalerS.fit_transform(input_dataset)

train_predict_shuff=model.predict(tdata_shuff).flatten()

# Erro
Err = mean_squared_error(tlabels,
train_predict_shuff.reshape(train_predict_shuff.shape[0],1))

```

```

Err1=np.sqrt(Err)

Matrixconectividade[i,idx-1] = sqrt(mean_squared_error(tlabels,
train_predict_shuff))
    #print(Matrixconectividade)

train_predict_shuff1=train_predict_shuff.reshape(train_predict_shuff.shape
[0],1)
    train_predict_shuff = scalerM.inverse_transform(train_predict_shuff1)


plt.figure(figsize=(6,4))
plt.title('Producer - '+str(idx), fontdict=font)
plt.plot(train_predict_shuff,c='g',linestyle='-',label='Shuffling I-
'+str(i+1),linewidth=1.5, markersize=1)
    #plt.plot(output_dataset,c='b',label='Real ',linewidth=1.5,
markersize=1)

plt.plot(train_predict,c='r',linestyle='dashed',label='Pred',linewidth=1.5
, markersize=1)
    plt.plot(output_dataset,c='b',label='Real',linewidth=1.5,
markersize=1)
    plt.xlabel('Time (days)',fontdict=font)
    plt.ylabel('Liquid Rate (bbl/day)',fontdict=font)
    plt.rcParams['xtick.labelsize'] = 20
    plt.rcParams['ytick.labelsize'] = 20
    plt.ylim([500, 2000])
    plt.xlim([0, 3000])
    plt.xticks(fontsize=15)
    plt.yticks(fontsize=15)
    plt.grid()

```

```

plt.legend(fontsize=16)
plt.show()

''' Matrix pre-connectivity'''

#Matrixconectividad[:,idx-1]=rms
idx=idx+1

#print(Matrixconectividad.shape)

# 5.- Soma de conectividades

Cwell=np.array(np.sum(Matrixconectividad,axis=1))
#print(Matrixconectividad)

Conectivity=np.zeros((ninj,nprod))
for i in range(ninj):
    for j in range(nprod):
        Conectivity[i-1,j-1]=Matrixconectividad[i-1,j-1]/Cwell[i-1]

print('Final Connectivity Values by RMS')
print(Conectivity)

```

ANEXO C – ANEXO DO NOTEBOOK DO MÉTODO DO GRADIENTE OU CNN1D

```

import os
import random
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from scipy.ndimage import zoom
from sklearn.preprocessing import StandardScaler
!pip install --upgrade keras-hypetune
from kerashypetune import KerasGridSearch
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.layers import *
from tensorflow.keras.models import Model
from tensorflow.keras.utils import plot_model
from tensorflow.keras.callbacks import *
from tensorflow.keras.optimizers import *
from tensorflow.keras import backend as K
from kerashypetune import KerasGridSearch
from pandas.plotting import register_matplotlib_converters
register_matplotlib_converters()
import math
from random import seed

#from keras.models import Model
#from keras.utils import plot_model
#''' numero de injetores'''

ninj=5

#''' numero de produtores'''

nprod=7

```

```
# ''' Define sequence length'''
```

```
sequence_length = 64
```

```
# Define pool size
```

```
pool_size=5
```

```
# PREDICTIONS vs REALITY ON TEST ###
```

```
lista=[100,300,500]
```

```
''' Well at analize on dataset'''
```

```
#''' Well inputs and output (at analize) '''
```

```
columns = ['I-1','I-2','I-3','I-4','I-5']
```

```
wells=[[ 'P-1'],[ 'P-2'],[ 'P-3'],[ 'P-4'],[ 'P-5'],[ 'P-6'],[ 'P-7']]
```

```
# ''' 1.- Load the dataset to train'''
```

```
dataset=pd.read_csv('/content/drive/MyDrive/Colab  
Notebooks/BCODataSet.dat',delimiter='\s+')
```

```
M=np.array(dataset)
```

```
M.shape
```

```
T=M[:,1:6]
```

```
T
```

```
font = {'family': 'serif','color':'darkblue','weight':'normal','size': 18}
```

```
time = np.arange(0, 3600, 500)
```

```

# demonstrate prediction
#for i in range(saida.shape[0]):
plt.figure(figsize=(6,4))
#plt.plot(time, T[:,0],color='blue', linewidth=2, linestyle='--',label='P-1')
plt.plot(T[:,0],color='blue', linewidth=2,label='I-1')
plt.plot(T[:,1],color='black', linewidth=2,label='I-2')
plt.plot(T[:,2],color='green', linewidth=2,label='I-3')
plt.plot(T[:,3],color='red', linewidth=2,label='I-4')
plt.plot(T[:,4],color='blue', linestyle='--', linewidth=2,label='I-5')

plt.legend(loc='upper right', shadow=True, fontsize='x-large')
plt.title('Injection Rate',fontdict=font)
plt.xlabel('Time (days)',fontdict=font)
plt.ylabel('bbl/day',fontdict=font)
plt.xlim([0, 4000])
#plt.ylim([1500, 2500])
plt.rcParams['xtick.labelsize'] = 20
plt.rcParams['ytick.labelsize'] = 20
plt.show()

```

```

# Matriz nula de conectividades

```

```

Matrixconectividade1=np.zeros((ninj,nprod))
Matrixconectividade2=np.zeros((ninj,nprod))
idx=1

```

```

for w in wells:

```

```

# ''' TRAIN, VALIDATION AND TEST SPLIT '''

```

```

indx1=int(math.floor(dataset.shape[0]*0.7))
indx2=int(math.floor(dataset.shape[0]*0.9))

# ''' TRAIN '''

y_train = dataset[dataset.day <= indx1][w].copy().values
train_date = dataset[dataset.day<= indx1].day.values
x_train = dataset[dataset.day <= indx1][columns].copy()

# ''' VALIDATION '''

y_val = dataset[indx1+1:indx2][w].copy().values
val_date = dataset[indx1+1:indx2].day.values
x_val = dataset[indx1+1:indx2][columns].copy()

# ''' TEST '''

y_test = dataset[indx2+1:dataset.shape[0]][w].copy().values
test_date = dataset[indx2+1:dataset.shape[0]].day.values
x_test = dataset[indx2+1:dataset.shape[0]][columns].copy()

# ''' PLOT TRAIN, VALIDATION, TEST '''

plt.figure(figsize=(6,4))
plt.plot(train_date, y_train, label='train', c='blue')
plt.plot(val_date, y_val, label='val', c='cyan')
plt.plot(test_date, y_test, label='test', c='orange')
plt.ylabel('Liquid Rate (bbl) ')
plt.xlabel('Time(days)')
plt.legend()
plt.show()

```

```

# ''' UTILITY FUNCTION FOR SEQUENCES GENERATION '''

def gen_sequence(id_df, seq_length, seq_cols):
    data_matrix = id_df[seq_cols].values
    num_elements = data_matrix.shape[0]

    for start, stop in zip(range(0, num_elements-seq_length),
range(seq_length, num_elements)):
        yield data_matrix[start:stop, :]

# ''' GENERATE SEQUENCES '''

X_train, X_val, X_test = [], [], []

for sequence in gen_sequence(x_train, sequence_length, columns):
    X_train.append(sequence)
for sequence in gen_sequence(x_val, sequence_length, columns):
    X_val.append(sequence)
for sequence in gen_sequence(x_test, sequence_length, columns):
    X_test.append(sequence)
X_train, X_val, X_test = np.asarray(X_train), np.asarray(X_val),
np.asarray(X_test)

# ''' UTILITY FUNCTIONS FOR TARGET SCALING '''

def scale_target(y, mean, std):
    return (y - mean)/std

def reverse_target(pred, mean, std):
    return pred*std + mean

# ''' SCALE TARGET '''

```



```

mean_train = y_train.mean()
std_train = y_train.std()

y_train_seq = scale_target(y_train[sequence_length:], mean_train,
std_train)
y_val_seq = scale_target(y_val[sequence_length:], mean_train, std_train)
y_test_seq = scale_target(y_test[sequence_length:], mean_train,
std_train)

# ''' SCALE SEQUENCES '''

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train.reshape(-
1,len(columns))).reshape(X_train.shape)
X_val = scaler.transform(X_val.reshape(-
1,len(columns))).reshape(X_val.shape)
X_test = scaler.transform(X_test.reshape(-
1,len(columns))).reshape(X_test.shape)

n_steps_in= X_train.shape[1]
n_features_in = X_train.shape[2]
n_steps_out=y_train_seq.shape[1]

### DEFINE MODEL ###

def set_seed(seed):
    tf.random.set_seed(seed)
    os.environ['PYTHONHASHSEED'] = str(seed)
    np.random.seed(seed)
    random.seed(seed)

inputs = Input(shape=(n_steps_in, n_features_in))
layer1 = Conv1D(filters=64, kernel_size=50, activation='relu',
padding='same', name='extractor')(inputs)

```

```

layer2 = MaxPooling1D(pool_size=2)(layer1)
layer3 = Flatten()(layer2)
layer4 = Dense(200,activation='relu')(layer3)
outputs = Dense(n_steps_out)(layer4)
model = Model(inputs=inputs, outputs=outputs)
model.compile(optimizer='adam', loss='mse')

monitor = EarlyStopping(monitor='val_loss',
                        min_delta=1e-4,
                        patience=50,
                        verbose=2,
                        mode='auto',
                        restore_best_weights=True)

history = model.fit(X_train, y_train_seq,
                    validation_data=(X_val, y_val_seq),
                    callbacks=[monitor],
                    verbose=1,
                    batch_size=64,
                    epochs=1000,
                    validation_split=0.2)

# ''' MAKE PREDICTION ON TEST '''
pred = reverse_target(model.predict([X_test]).ravel(), mean_train,
std_train)

print(model.summary())
plot_model(model, show_shapes=True)

for i in lista:
    print(i)
    strg = 'Sequence length = ' +str(sequence_length)
    plt.figure(figsize=(16,6))

```

```

    start= 1
    end= i
    plt.plot(test_date[sequence_length+start:end+sequence_length],
y_test[sequence_length+start:end+sequence_length],c='orange', label='real
test')
    plt.plot(test_date[sequence_length+start:end+sequence_length],
pred[start:end], c='green', label='prediction')
    #plt.ylim([1500, 3500])  ***I want to keep these values fixed"
    plt.legend();
    plt.title(strg)
    plt.ylabel('Rate')
    plt.show()

```

UTILITY FUNCTIONS FOR GRADIENT IMPORTANCE AND ACTIVATION MAPS
GENERATION ###

```

def gradient_importance(seq, model):
    seq = tf.Variable(seq[np.newaxis,:,:], dtype=tf.float32)
    with tf.GradientTape() as tape:
        predictions = model(seq)
    grads = tape.gradient(predictions, seq)
    grads = tf.reduce_mean(grads, axis=1).numpy()[0]
    return grads

def activation_grad(seq, model):
    seq = seq[np.newaxis,:,:]
    grad_model =
Model([model.inputs],[model.get_layer('extractor').output,model.output])
    # Obtain the predicted value and the intermediate filters
    with tf.GradientTape() as tape:
        seq_outputs, predictions = grad_model(seq)

    # Extract filters and gradients
    output = seq_outputs[0]

```

```

grads = tape.gradient(predictions, seq_outputs)[0]

# Average gradients spatially
weights = tf.reduce_mean(grads, axis=0)

# Get a ponderated map of filters according to grad importance
cam = np.ones(output.shape[0], dtype=np.float32)
for index, w in enumerate(weights):
    cam += w * output[:, index]
    time = int(seq.shape[1]/output.shape[0])
    cam = zoom(cam.numpy(), time, order=1)
    heatmap = (cam - cam.min())/(cam.max() - cam.min())
    return heatmap

### DISPLAY INPUT SEQUENCE ###

id_ = 64
plt.figure(figsize=(6,4))
plt.plot(X_test[id_])
plt.ylabel('std series'); plt.xlabel('time lags'); plt.legend(columns)
plt.show()
#print('X_test.shape = ', X_test.shape)
seq_=X_test[id_]
#print('seq_.shape = ', seq_.shape)
seq1_=seq_[np.newaxis,:,:]
#print('seq1_.shape = ', seq1_.shape)

### GRADIENTS IMPORTANCE ###
grad_imp1 = gradient_importance(X_test[id_], model)
strg = 'RELATIVE = ' +str(sequence_length)
plt.figure(figsize=(6,4))
plt.bar(range(len(grad_imp1)), grad_imp1)
plt.xticks(range(len(columns)), columns, rotation=90)
plt.ylabel('gradients');

```

```

plt.title(strg)
plt.show()
#print('values1')
#print(grad_imp1)

# Original values connectivity in function gradient values
for i in range(ninj):
    Matrixconectividade1[i,idx-1]=grad_imp1[i]
    #print(Matrixconectividade1)

def abs_gradient_importance(seq, model):
    seq = tf.Variable(seq[np.newaxis,:,:], dtype=tf.float32)
    with tf.GradientTape() as tape:
        predictions = model(seq)
    grads = tape.gradient(predictions, seq)
    grads = tf.reduce_mean(tf.abs(grads), axis=1).numpy()[0]
    return grads

def test_gradient_importance(seq, model):
    seq = tf.Variable(seq[np.newaxis,:,:], dtype=tf.float32)
    with tf.GradientTape() as tape:
        predictions = model(seq)
    grads = tape.gradient(predictions, seq)
    return grads

# VALUES CONNECTIVITIES
#print('values2')
grad_test = test_gradient_importance(X_test[id_], model)

### ABSOLUTE GRADIENTS IMPORTANCE ###
grad_imp2= abs_gradient_importance(X_test[id_], model)

```

```

for i in range(ninj):
    Matrixconectividade2[i,idx-1]=grad_imp2[i]
    #print(Matrixconectividade2)

strg = 'ABSOLUTE = ' +str(sequence_length)
plt.figure(figsize=(6,4))
plt.bar(range(len(grad_imp2)), grad_imp2)
plt.xticks(range(len(columns)), columns, rotation=90)
plt.ylabel('gradients');
plt.title(strg)
plt.show()
#print(grad_imp2)

### GENERATE ACTIVATION ON INPUT SEQUENCE ###
activation = activation_grad(X_test[id_], model)
plt.figure(figsize=(6,4))
plt.plot(X_test[id_])
plt.ylabel('std series');
plt.xlabel('time lags');
plt.twinx()
plt.imshow(np.vstack([activation]*30), alpha=0.35)
plt.axis('off'); plt.colorbar()
plt.show()
idx=idx+1

# 5.- Soma de conectividades

Cwell1=np.array(np.sum(Matrixconectividade1,axis=1))
Cwell2=np.array(np.sum(Matrixconectividade2,axis=1))

# Connectivity relative

```

```
CR=np.zeros((ninj,nprod))
for i in range(ninj):
    for j in range(nprod):

        CR[i,j]=Matrixconectividade1[i,j]*(1/Cwell1[i])
```

```
# Connectivity Absolute
```

```
CA=np.zeros((ninj,nprod))
for i in range(ninj):
    for j in range(nprod):
        CA[i,j]=Matrixconectividade2[i,j]*(1/Cwell2[i])
```