UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
ESCOLA DE ENGENHARIA
ENG. DE CONTROLE E AUTOMAÇÃO

JONAS HALMENSCHLAGER

COMPARAÇÃO DE REDES NEURONAIS RECORRENTES CONTÍNUA E DISCRETA PARA A MODELAGEM CAIXA-PRETA DE UM SISTEMA DE ESCOAMENTO DE PETRÓLEO

Porto Alegre
2019

JONAS HALMENSCHLAGER

COMPARAÇÃO DE REDES NEURONAIS RECORRENTES CONTÍNUA E DISCRETA PARA A MODELAGEM CAIXA-PRETA DE UM SISTEMA DE ESCOAMENTO DE PETRÓLEO

Trabalho de Conclusão de Curso (TCC-CCA) apresentado à COMGRAD-CCA da Universidade Federal do Rio Grande do Sul como parte dos requisitos para a obtenção do título de *Bacharel em Eng. de Controle e Automação* .

ORIENTADOR: Prof. Dr. Pedro Rafael Bolognese Fernandes

Porto Alegre
2019

JONAS HALMENSCHLAGER

COMPARAÇÃO DE REDES NEURONAIS RECORRENTES CONTÍNUA E DISCRETA PARA A MODELAGEM CAIXA-PRETA DE UM SISTEMA DE ESCOAMENTO DE PETRÓLEO

Este Trabalho de Conclusão de Curso foi julgado adequado para a obtenção dos créditos da Disciplina de TCC do curso *Eng. de Controle e Automação* e aprovado em sua forma final pelo Orientador e pela Banca Examinadora.

Orientador: _____
Prof. Dr. Pedro Rafael Bolognese Fernandes, UFRGS
Doutor pela pela TU Dortmund – Dortmund, Alemanha

Banca Examinadora:

Prof. Dr. Marcelo Götz, UFRGS
Doutor pela Universitat Paderborn – Paderborn, Alemanha

Prof. Dr. Renato Ventura Bayan Henriques, UFRGS
Doutor pela UFMG – Belo Horizonte, Brasil

_____
Atual Coordenador do Curso
Coordenador de curso
Eng. de Controle e Automação

Porto Alegre, julho de 2019.

DEDICATÓRIA

Dedico este trabalho à meus pais, Clarice e Vanderlei, que nunca deixaram de acreditar em meu potencial.

AGRADECIMENTOS

À Universidade Federal do Rio Grande do Sul (UFRGS) pela oportunidade de realização dos estudos e ao povo brasileiro pelo suporte financeiro da instituição.

Ao meu prezado orientador Pedro pela dedicação em tornar este trabalho o melhor possível, permitir uma troca de conhecimento inigualável e ser um docente diferenciado.

Aos meus colegas de curso que me auxiliaram durante a graduação e durante o trabalho, em especial ao Carlos Oliveira por oferecer sua máquina para simulação dos modelos.

À minha namorada Sabrina pelo apoio motivacional.

Aos meus colegas do Centro Estudantes Universitários de Engenharia (CEUE) por fazerem da Universidade um lugar melhor.

RESUMO

A modelagem de sistemas dinâmicos é uma área de grande importância para o controle e a otimização de processos. A modelagem não-linear é particularmente desafiante, uma vez que sistemas deste tipo podem apresentar estruturas arbitrárias. Deste modo, os modelos empregados devem apresentar flexibilidade para modelar diferentes tipos de comportamentos dinâmicos. Redes Neuronais são bastante empregadas em modelagem estacionária por permitirem ajustar qualquer mapeamento estático de entrada e saída se os dados forem suficientes para o ajuste. Este trabalho de conclusão de curso compara e analisa os resultados obtidos na modelagem caixa-preta de um sistema dinâmico não-linear por meio de redes neuronais recorrentes em domínio de tempo discreto e contínuo, em que a rede contínua é treinada para o termo de taxa de variação $f(x)$ do sistema dinâmico em espaço de estados $f(x) = \dot{x}$ e é necessária uma etapa intermediária de integração do modelo neuronal obtido. Com isto, pretende-se evitar as dificuldades de determinação da estrutura e de ajuste (viés) das redes dinâmicas do tipo recorrente. Os resultados obtidos com os modelos neuronais recorrentes também são comparados com o método de regressão linear.

Palavras-chave: Rede Neuronal, Modelagem de Sistemas, Inteligência Artificial.

ABSTRACT

The modeling of dynamic systems is an important area for the process control and optimization. Nonlinear modeling is particularly a challenge, once there are systems of arbitrary structures. In this way, a model used must be flexible enough to model different types of dynamic behaviors. Neural Networks are widely used in stationary modeling because they allow to adjust any static input and output mapping if the data is enough for the optimization. This paper compare and analyse the neural networks results achieved in a nonlinear and dynamical system black-box modeling, where the continuos time network is trained for the rate of change term $f(x)$ of the dynamic system in state space $f(x) = \dot{x}$ and is required an intermediate stage of integration for this obtained neuronal model. With this, it is intended to avoid the difficulties of determining the structure and adjustment (*bias*) of the dynamic networks of the recurrent type. The results also will be compared with linear regression method.

Keywords: Artificial Neural Network, System Modeling, Deep Learning.

SUMÁRIO

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS

RNA	Rede Neuronal Artificial
PID	Proporcional Integral Derivativo
MCP	McCulloch-Pitts
NFIR	Nonlinear Finite Impulse Response
NARX	*Nonlinear Autoregressive model process with eXogenous input*
NOE	*Nonlinear Output Error*
NARMAX	*Nonlinear AutoRegressive Moving Average with eXogenous inputs*
NBJ	*Nonlinear Box-Jenkins*
ADALINE	*ADAptive LInear NEuron*
RNAR	Rede Neuronal Artificial Recorrente
SNL	Sistema Não-Linear
MLS	*Maximum-Length-Sequence*
TFFT	TransFormada de Fourier
WT	*Wavelet Transform*
ANFIS	*Adaptive Neural Fuzzy Inference Systems*
EDO	Equação Diferencial Ordinária
MSE	*Mean Square Error*
ANM	Árvore de Natal Molhada
FOWM	*Fast Offshore Wells Model*
MIMO	*Multiple Input, Multiple Output*
MISO	*Multiple Input, Single Output*

LISTA DE SÍMBOLOS

$\sum$	Somatório
ϕ	Função de Ativação
j	Número de Neurônios
L	Número de Atrasos
w_j	Peso Sináptico do Neurônio j
b_j	Peso de *bias* do Neurônio j
I	Matriz Identidade
E	Erro Total
e	Erro
J	Matriz Jacobiana
μ	Coeficiente de Combinação
u	Sinal de Entrada
y	Sinal de Saída
x	Estado do Sistema
$\hat{y}$	Sinal de Saída Estimado
t	Tempo
R	Coeficiente de Correlação
Ps	Pressão do Separador
$Choke$	Abertura da Válvula Choke
Gl	Vazão do Gás *lift*
Prt	Pressão de Topo do *Ryser*
Prb	Pressão na Base do *Ryser*
Ptt	Pressão do Tipo do Tubing
$Ppdg$	Pressão Medida no Olho do Poço

1 INTRODUÇÃO

Os sistemas nas áreas de conhecimento da engenharia podem ser representados por modelos matemáticos baseados em conjuntos de equações correspondentes a leis da Física. O conjunto de equações objetiva representar uma aproximação do sistema real observado na tentativa de explicar e prever o seu comportamento quando submetido à diferentes excitações.

A inclusão de todas as características de um sistema real em seu modelo matemático é geralmente inviável, sendo comumente imposto simplificações no processo de modelagem, por exemplo, como premissas de não-linearidade ou invariância no tempo dentro uma determinada faixa de operação. No entanto, estas hipóteses, quando válidas, servem apenas uma estreita faixa de operação.

A obtenção das características de um sistema não-linear e dinâmico sem o conhecimento prévio de seus princípios físicos é possível através da modelagem empírica ou *caixa-preta*, em que apenas o acesso aos dados de entrada e saída é necessário (SJÖBERG et al., 1995). Existem diversos métodos de modelagem não-linear caixa-preta: *Wavelet Transform* (WT), modelos *fuzzy*, *Adaptive Neural Fuzzy Inference Systems* (ANFIS), e as redes neuronais artificiais (RNAs). Este conceito pode ser desenvolvido no domínio de tempo discreto ou contínuo. A determinação da estrutura e o ajuste do modelo são etapas necessárias no processo de identificação de tempo discreto e geralmente podem dificultar o processo. Uma dificuldade em particular na obtenção de modelos não-lineares discretos para processos contínuos é que, ao contrário do caso linear, não há uma forma de discretização que preserve a estrutura do sistema, tornando esta escolha mais complexa.

A modelagem de um sistema dinâmico não-linear pode ser medida em termos de quatro critérios segundo (R. PEARSON, 2003): precisão de aproximação; interpretação física; adEquação a controle e facilidade no desenvolvimento. Modelos lineares de baixa ordem geralmente atendem aos dois últimos critérios, porém são incapazes de representar fenômenos importantes. Outro fator a ser considerado é o domínio do tempo no qual o sistema é representado, pois quando discreto, um modelo não-linear têm os dois primeiros critérios comprometidos devido à perda de informação na escolha da sua estrutura, enquanto que em tempo contínuo esta etapa não é necessária (R. PEARSON, 2003).

Neste trabalho é testada a identificação de sistemas dinâmicos não-lineares contínuos através de um modelo RNA dinâmico no domínio de tempo contínuo. Para tanto, dados de entrada e saída são utilizados na etapa de treinamento para o termo de taxa de variação $f(x)$ do sistema dinâmico ($f(x) = \dot{x}$). Para o treinamento, o modelo proposto é integrado numericamente para a geração das saídas. O método proposto é validado por meio de métricas de desempenho e os resultados obtidos são comparados com estruturas de identificação discretas.

1.1 Extração de Petróleo

Profundidades acima de milhares de metros são facilmente atingidas por plataformas plataformas *offshore* de extração de petróleo marinho. Misturas formadas por gás, água e o próprio petróleo são extraídas e transportados através dos poços para a superfície, como mostra a Figura 1. Na superfície encontra-se a Árvore de Natal Molhada (ANM), onde estão dispostos os equipamentos de instrumentação e a válvula de estrangulamento que controla o fluxo do fluido, a qual é acionada remotamente. Na *flow line* (ou linha de produção), as produção de todos os poços são combinadas e então elevadas para a plataforma através de dutos verticais, ou *risers*, onde ocorre o processo de separação das fases do fluido e é onde se encontra a válvula *Choke* responsável por garantir o funcionamento dos equipamentos da superfície (DI MEGLIO et al., 2012).

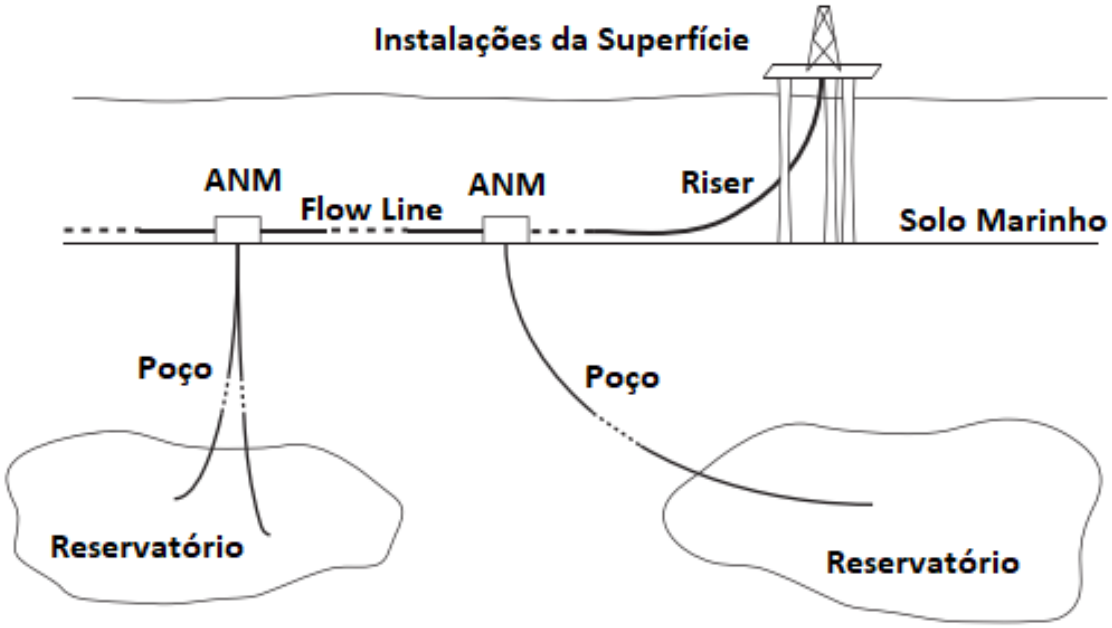

Figura 1: Plataforma de Extração de Petróleo *offshore*
Fonte: Adaptado de (DI MEGLIO et al., 2012)

A distribuição não-homogênea do petróleo na forma gasosa e líquida pode provocar bolhas de gás nas tubulações prejudiciais à produção e de alto risco à operação (SCHMIDT; BRILL; BEGGS, 1979). Aliadas à enorme pressão causada pela profundidade de fluido acumulado ao longo do poço, a presença de eventuais curvas no poço e de inclinações acentuadas no *riser* podem bloquear a passagem do fluxo de gás. Ao ser bloqueado, o gás forma uma bolha comprimida que, ao vencer a pressão hidrostática causada pela coluna de fluido, expande empurrando o líquido impulsivamente, causando as denominadas golfadas que caracterizam a vazão produzida como oscilatória e tornam o processo instável (DI MEGLIO et al., 2012).

O sistema dinâmico não-linear como objeto de estudo é um modelo semi-empírico que simula a extração de petróleo em plataforma marinha *offshore*, o FOWM. O comportamento não-linear presente na resposta do sistema ocorre devido ao fenômeno de golfadas, o qual é comum em processos de extração de petróleo e significadamente prejudicial à operação. O emprego da modelagem caixa-preta através de redes neuronais é útil na previsão das não-linearidades do sistema.

1.2 Objetivo

Este trabalho tem como objetivo comparar e analisar Redes Neuronais Artificiais Recorrentes (RNARs) no domínio do tempo discreto e contínuo para a modelagem de sistemas. A comparação se dá principalmente em termos da estrutura das redes resultantes em relação à capacidade de predição. Para esta comparação, foram usados dados gerados através de um modelo semi-empírico dinâmico não-linear, o FOWM, em espaço de estados $f(x) = \dot{x}$.

1.3 Estrutura do Trabalho

Este trabalho de conclusão de curso está organizado da seguinte forma: O Capítulo 2 compreende a revisão bibliográfica dos conceitos de modelagem de sistemas dinâmicos, de redes neuronais e de excitação de sistemas SNLs. No Capítulo 3 são apresentados os materiais e os métodos utilizados no trabalho. No Capítulo 4 são mostrados os resultados obtidos para diferentes combinações de arquitetura de rede discreta, onde a rede de melhor desempenho é selecionada para espelhar um modelo de tempo contínuo. No Capítulo 5, finalmente, o trabalho tem a avaliação dos resultados alcançados em referência ao objetivo, conclusões finais e proposta de trabalhos futuros.

2 REVISÃO BIBLIOGRÁFICA

Nas seções a seguir são descritas as referências bibliográficas utilizadas neste trabalho de conclusão de curso.

2.1 Modelagem de Sistemas Dinâmicos

Existem basicamente duas abordagens para modelar um sistema dinâmico: pela física do processo ou por identificação de sistemas. A dificuldade na modelagem pela física é que geralmente não se tem o conhecimento dos fundamentos físicos que regem o processo, sendo necessária a segunda abordagem, a qual pode ser realizada por dois diferentes tipos de abordagens (LJUNG; SÖDERSTRÖM, 1983):

- Métodos paramétricos;
- Métodos não-paramétricos.

A utilização de séries temporais permite a construção de modelos de processos estocásticos que buscam estimar o valor futuro da variável de um sistema com base nos seus valores passados somente, em que o sistema apresenta um padrão persistente ou sistemático no comportamento da variável a ser estimada através da representação paramétrica (PINDYCK; RUBINFELD, 1998).

Nos métodos paramétricos o problema central é estimar um vetor de parametrização finito utilizado na identificação do melhor modelo que descreva o comportamento do sistema, em que o vetor é estimado a partir da observação dos dados de entrada e saída do sistema. Já nos métodos não-paramétricos não há o emprego de um número finito de parâmetros (LJUNG; SÖDERSTRÖM, 1983).

2.1.1 Tempo Discreto

Nos modelos entrada/saída se tem a descrição da resposta em função da entrada aplicada, sendo que os princípios físicos que regem o funcionamento do sistema são negligenciados (SJÖBERG et al., 1995). O problema da modelagem caixa-preta de um sistema dinâmico passa a ser mais complexo quando características não-lineares devem ser consideradas. A partir da observação dos valores de passado de entrada e saída e da estrutura não-linear da Equação 1, é possível estimar o valor de saída $y(t)$ dada uma entrada $u(t)$ (SJÖBERG et al., 1995).

$$y(t) = g(u^{(t-1)}, y^{(t-1)}) + v(t) \tag{1}$$

Na Equação 1, a presença do termo $v(t)$ significa que a próxima saída $y(t)$ não será exatamente a saída prevista. Desta forma, o objetivo durante a identificação do sistema dinâmico discreto através da Equação 1 é obter $g(u^{t-1}, y^{t-1})$ dentro de uma família de funções de modo a resultar numa boa aproximação da saída observada, a despeito do ruído $v(t)$. No caso das abordagens paramétricas, esta função será parametrizada pelo vetor de dimensão finita θ, resultando em $g(u^{(t-1)}, y^{(t-1)}, \theta)$ (SJÖBERG et al., 1995).

O mapeamento dos valores observados da entrada e da saída no passado é denominado por vetor de variáveis de regressão e é representado por ϕ, enquanto a predição da saída $\hat{y}(t)$ é dada pela Equação 2. Segundo (SJÖBERG et al., 1995),a modelagem caixa-preta de sistemas dinâmicos não-lineares, portanto, requer duas etapas: a seleção de uma estrutura funcional $g(\phi)$ parametrizada por θ e a escolha do vetor de variáveis de regressão ϕ:

$$\hat{y}(t|\theta) = g(\phi(t), \theta) \tag{2}$$

A viabilidade da aplicação de redes neuronais na identificação de sistemas de tempo discreto tem sido levantadas em divesos estudos (CHEN; S. BILLINGS; GRANT, 1990; CHEN; S. BILLINGS, 1992). Diferentes modelos podem ser derivados para cada escolha de vetor de regressão. O modelo auto-regressivo não-linear com entradas exógenas (LEONTARITIS; S. A. BILLINGS, 1985) *Nonlinear Autoregressive model process with eXogenous input* (NARX)), por exemplo, é formado por valores de presente e passado da entrada e da saída disponíveis, representando entradas exógenas (externas ao sistema). Seu vetor é dado pela Equação 3, onde $k_y + k_u$ é o número de regressores utilizados no modelo.

$$\phi(t) = \left[\begin{array}{cccccc} y(t-1) & \ldots & y(t-k_y) & u(t-1) & \ldots & u(t-k_u) \end{array} \right]^T \tag{3}$$

Diferentemente do NARX, os modelos *Nonlinear Output Error* (NOE), *Nonlinear AutoRegressive Moving Average with eXogenous inputs* (NARMAX) e (*Nonlinear Box-Jenkins* (NBJ) são derivados de vetores de variáveis de regressão que dependem de valores passados do modelo parametrizado $\hat{y}(t-k|\theta)$ (LEONTARITIS; S. A. BILLINGS, 1985). A Tabela 1 mostra os valores de ϕ para cada modelo, onde $\hat{y}_u$ e ϵ_u são os valores simulados da saída e do erro entre a saída estimada e a saída real respectivamente.

Tabela 1: Modelos derivados para cada vetor de regressão

NOE	$\hat{y}_u(t-k\|\theta)$	$u(t-k)$		
NARMAX	$y(t-k)$	$u(t-k)$	$\epsilon(t-k\|\theta)$	
NBJ	$\hat{y}_u(t-k\|\theta)$	$u(t-k)$	$\epsilon(t-k\|\theta)$	$\epsilon_u(t-k\|\theta)$

2.1.2 Tempo Contínuo

Quanto à representação, um sistema pode ter seu modelo ou em espaço de estados ou na forma de entrada/saída. Sistemas físicos identificados geralmente não tem relação direta com os parâmetros de um modelo discreto, dificultando sua interpretação física (UNBEHAUEN; RAO, 1990). Segundo (R. PEARSON, 2003), em casos não-lineares, a

estrutura dificilmente é preservada mesmo que a discretização seja exata. Em sua maioria, os sistemas estudados no campo da engenharia são de tempo contínuo e suas modelagens em tempo contínuo não necessitam a definição de um vetor de variáveis de regressão. Publicações sobre modelos paramétricos utilizados na identificação de sistemas dinâmicos contínuos tem recebido bastante atenção (AGUIRRE; S. A. BILLINGS, 1995; LJUNG, 1999; UNBEHAUEN; RAO, 1990; R. PEARSON, 2003).

Existem basicamente duas formas de representar um sistemas em domínio de tempo contínuo, ou por função de transferência ou por espaço de estados. Uma função de transferência simula o comportamento de saída do sistema para determinado estímulo de entrada através da relação $\hat{y}(t,\theta) = G(\epsilon,\theta) \cdot u(t)$, em que ϵ é um operador linear qualquer, θ é um vetor de parâmetros e G é dada pela Equação 4.

$$G(\epsilon,\theta) = \frac{B(\epsilon,\theta_a)}{A(\epsilon,\theta_b)} = \frac{b_0 + b_1\epsilon + ... + b_m\epsilon^m}{1 + a_1\epsilon + a_2\epsilon^2 + ... + a_n\epsilon^n} \tag{4}$$

Em espaço de estados, há uma relação entre as variáveis de entrada, saída e de estados através de uma Equação diferencial (ou de diferenças, para o caso discreto) (LJUNG; SÖDERSTRÖM, 1983). Um sistema dinâmico pode ser representado em espaço de estados pelo sistema de equações 5. Segundo (OGATA, 1999), o espaço de estado é o espaço n-dimensional de em que os eixos coordenados são formados por n variáveis de estado do sistema.

$$\begin{cases} \frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u},t) \\ \mathbf{y} = \mathbf{g}(\mathbf{x},\mathbf{u},t) \end{cases} \tag{5}$$

A maior parte das pesquisas relacionadas à redes em tempo contínuo têm sido direcionadas à redes de arquitetura *Feedforward*. Algumas publicações tem tratado de redes recorrentes em tempo contínuo para sistemas invariantes no tempo: (FUNAHASHI; NAKAMURA, 1993) estuda a aplicação de uma rede de Hopfield (HOPFIELD, 1982) para um sistema sem variação da entrada $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x})$; (CHOW; LI, 2000) aplica modelos mais genéricos do tipo $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x},\mathbf{u})$ e (FUNAHASHI; NAKAMURA, 1993) considera um sistema do tipo $\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x}) \cdot \mathbf{u}$.

A seleção de uma estrutura razoável na modelagem de sistema não-lineares é uma tarefa difícil, visto que ela é baseada na experiência com modelos lineares de baixa ordem e pode falhar quando confrontada com sistemas não lineares (R. PEARSON, 2003). Também, a discretização de um modelo contínuo pode gerar perda de informação sobre alguns de seus fenômenos físicos e estar restrito para determinados valores de tempo de amostragem (R. PEARSON, 2003).

2.2 Redes Neuronais Artificiais

Os desenvolvimentos científicos na área de redes neuronais artificias tiveram como inspiração a capacidade de o cérebro humano aprender linguagens, classificar e interpretar imagens, entre outras formas de aprendizagem. A habilidade de aprender se deve à grande quantidade de neurônios e sinapses (também conhecidas como conexões) existentes no cérebro, podendo chegar a 10 bilhões e 60 trilhões respectivamente (SHEPHERD, 2003).

Um neurônio biológico é formado basicamente por dendritos, corpo celular, núcleo, axônio, ramificações e terminais do axônio. Seu funcionamento se dá por transformações

químicas complexas em que sinais sinápticos são transformados em sinais elétricos no corpo celular e transmitidos através do axônio e seus terminais para demais neurônios conectados pelas sinapses (SHEPHERD, 2003). A estrutura de um neurônio biológico é mostrada na Figura 2.

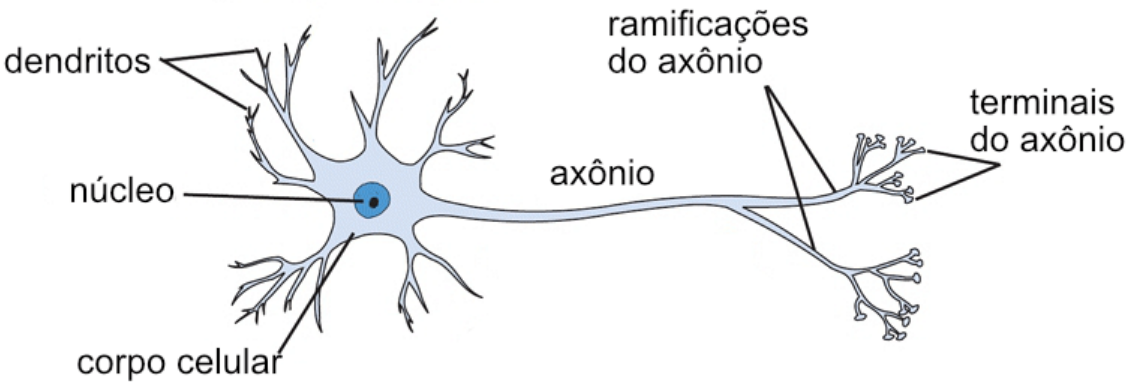

Figura 2: Neurônio Biológico
Fonte: Adaptado de (CS231N, 2019)

O primeiro trabalho dedicado ao estudo de um neurônio artificial foi publicado em (MCCULLOCH; PITTS, 1943). Neste trabalho foi apresentada a recriação artificial de um neurônio biológico, o que ficou conhecido como o Neurônio Booleano de McCulloch-Pitts (MCP). No entanto, ainda não são mencionadas técnicas de treinamento e problemas não-linearmente separáveis não poderiam ser resolvidos. Já o modelo de neurônio não-linear mostrado na Figura 3 é baseado no modelo de MCP, porém com o incremento de um *bias* e uma função de ativação diferente da *Heaviside*, é capaz de solucionar problemas mais complexos (HAYKIN, 2007).

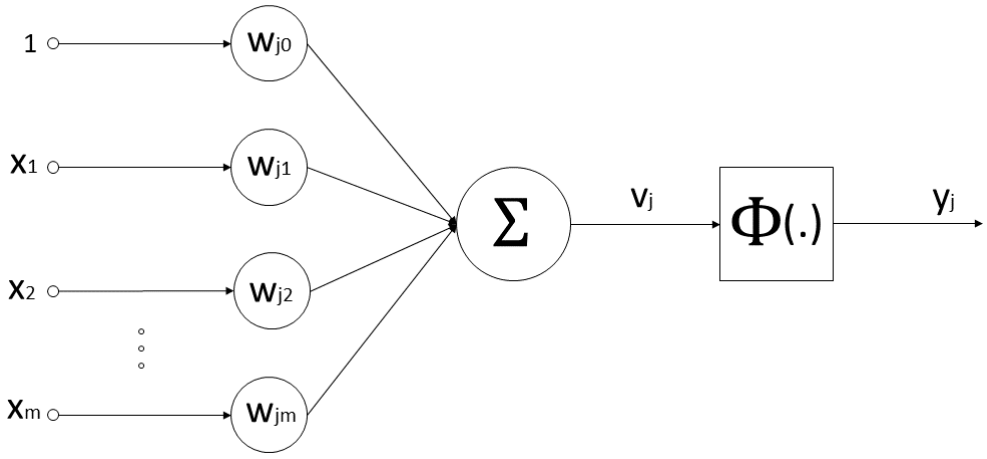

Figura 3: Modelo do Neurônio Artificial
Fonte: Adaptado de (HAYKIN, 2007)

O neurônio de índice j mostrado na Figura 3 modela um função matemática, a qual pondera os sinais de entrada $(x_1, x_2, ..., x_m)$ pelos pesos sinápticos $(w_{j1}, w_{j2}, ..., w_{jm})$ que, após combinados com o peso de *bias* w_{j0} no Somador e multiplicados pela função de ativação ϕ, forma o sinal de saída y_j conforme a Equação 6.

$$y_j = \phi(\sum_{i=1}^{m}(w_{ji}x_i) + w_{j0}) \tag{6}$$

2.2.1 Função de Ativação

Realiza uma transformação não-linear nos dados de entrada, tornando-o modelo capaz de aprender e executar tarefas mais complexas. Além disso, diferencia informações recebidas pelo neurônio que serão utilizadas/ativadas ou rejeitadas. São utilizados para converter grandes saídas das unidades em um valor menor, além de promover a não linearidade na rede

A função de ativação é responsável por limitar a amplitude do sinal y_j a um valor finito tipicamente contido no intervalo $[0, 1]$, *Heaviside* e Sigmóide, ou $[-1, 1]$, Linear por Partes e Tangente Hiperbólico. Além disto, elas são responsáveis por tornar o modelo neuronal capaz de aprender tarefas mais complexas, diferenciando as informações que deve ser ativadas ou rejeitadas (GÉRON, 2017). As principais funções de ativação são mostrados na Tabela 2.

Tabela 2: Principais Funções de Ativação

Nome da Função	Função
Heaviside	$\phi(v_j) = \begin{cases} 1, & \text{se } v_j \geq 0 \\ 0, & \text{se } v_j < 0 \end{cases}$
Linear por Partes	$\phi(v_j) = \begin{cases} 1, & \text{se } v_j > 0 \\ 0, & \text{se } v_j = 0 \\ -1, & \text{se } v_j < 0 \end{cases}$
Sigmóide	$\phi(v_j) = \frac{1}{1 + e^{-av_j}}$
Tangente Hiperbólico	$\phi(v_j) = \frac{e^{av_j} - e^{-av_j}}{e^{av_j} + e^{-av_j}}$

A função de ativação Sigmóide é a mais utilizada atualmente na construção de redes neuronais (HAYKIN, 2007). Ela possui algumas vantagens em relação às demais, como ser diferenciável, reproduzir sinais contínuos em sua saída e gerar uma característica não-linear. É importante destacar que o parâmetro a contido no denominador corresponde à inclinação da função e, quando este tende ao infinito, tem-se a alteração de comportamento da função de sigmóide para *Heaviside*, visto que suas respostas possíveis passam a ser 0 ou 1 (HAYKIN, 2007).

2.2.2 Arquitetura

Fazem parte da definição de arquitetura: o número de camadas da rede, camada única ou múltiplas camadas; a topologia, *Feedforward* ou recorrente; a conectividade entre os neurônios, forte ou fraca (PÁDUA BRAGA, 2000). As arquiteturas geralmente possuem entre duas e três camadas em sua arquitetura, como mostra a Figura 5. Cada camada possui as seguintes funções:

- Camada de Entrada: entrada dos dados;
- Camadas Oculta: processamento dos dados;

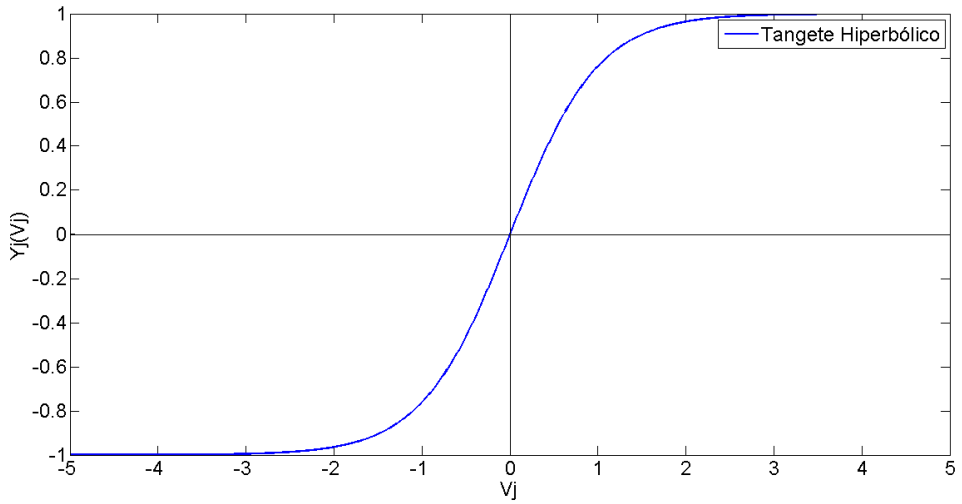

Figura 4: Função Tangente Hiperbólico
Fonte: Autor

- Camada de Saída: processamento dos dados e cálculo do resultado.

A arquitetura *Feedforward* com múltiplas camadas possui mais de uma camada de neurônios, sendo estas camadas adicionais ocultas. Esta cacarterística tem como objetivo fazer a interface entre a camada de entrada e a de saída, tornando a rede capaz de extrair características de ordem elevada de um sistema uma vez que o tamanho da camada de entrada pode ser maior (CHURCHLAND; SEJNOWSKI, 1992).

Redes de topologia recorrente tem se mostrado interessante em aplicações que envolvam a identificação de sistemas dinâmicos devido à suas capacidades de processamento temporal e de implementação de memórias adaptativas (BENGIO et al., 1993). Também, estas redes têm mostrados vantagens frente às redes de topologia *Feedforward* (HORNIK; STINCHCOMBE; WHITE, 1989).

O comportamento dinâmico não-linear de uma RNAR se deve à existência de pelo menos um laço de realimentação. Na Figura 6, por exemplo, há uma realimentação conectando o neurônio da camada de saída com a entrada de uma rede no domínio de tempo contínuo. Quando em tempo discreto, a realimentação envolve um atraso de sinal.

2.2.3 Treinamento

O processo de treinamento de uma RNA consiste em um processo de iterações sucessivas para o ajuste ótimo de seus parâmetros (PÁDUA BRAGA, 2000). Estes algoritmos são classificados em supervisionados ou não-supervisionados. Em ambas as classificações há a informação da entrada do sistema a ser representado, porém nos métodos supervisionados a saída real de um neurônio (que deve ser realimentada) é substituída pela resposta desejada correspondente, o *target* (HAYKIN, 2007). Ao final do processo, a rede adquire propriedades de comportamento do sistema pelo qual recebera estímulos durante seu aprendizado.

O algoritmo de treinamento *backpropagation error* é amplamente utilizado atualmente, entretanto, sabe-se que sua convergência é lenta em comparação à demais métodos de treinamento (YU; WILAMOWSKI, 2011). O algoritmo Levenberg–Marquardt (LEVENBERG, 1944; MARQUARDT, 1963) combina o método do gradiente descendente

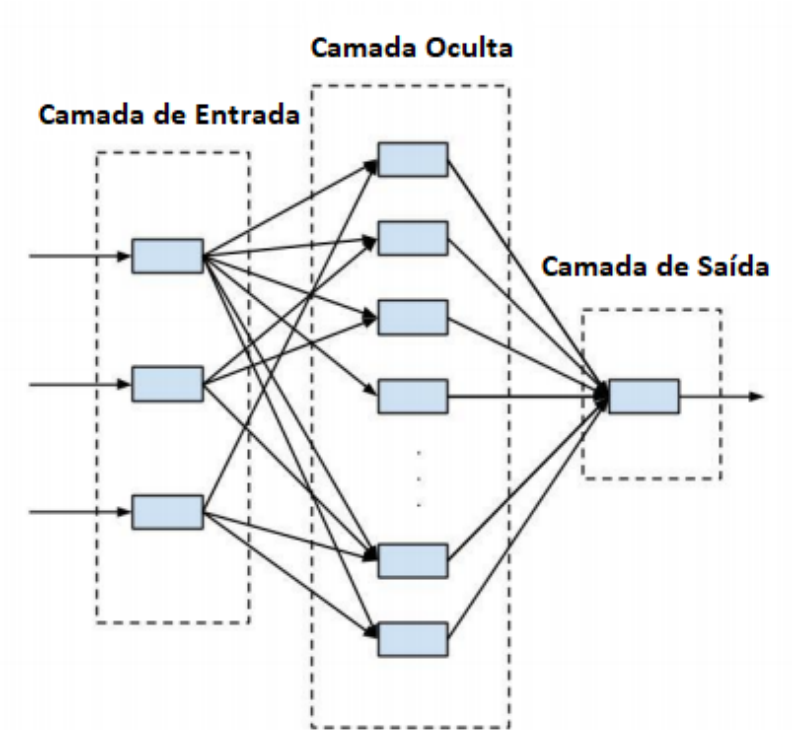

Figura 5: Disposição das Camadas de uma RNA
Fonte: Adaptado de (DEKA et al., 2016)

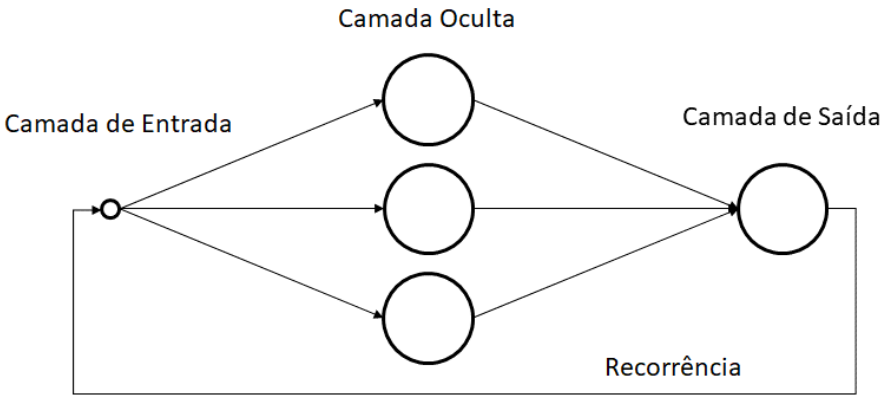

Figura 6: Rede Neuronal Artificial Recorrente Contínua
Fonte: Autor

com o algoritmo de Gauss-Newton, aplicando o primeiro em superfícies de curvatura complexa e o segundo quando se torna adequada uma aproximação quadrática (YU; WILAMOWSKI, 2011).

A primeira etapa do algoritmo de Levenberg–Marquardt consiste em calcular o erro total em função das entradas do neurônio e dos pesos sinápticos (inicialmente definidos de forma randômica). A Equação 7 mostra o cálculo do erro total para uma rede formada por j neurônios e somente 1 entrada, em que o erro é a diferença entre a saída desejada e a saída real de cada de cada neurônio, isto é, $e = \hat{y} - y$.

$$E(\mathbf{x}, \mathbf{w}) = \sum_{i=1}^{j} e_i^2 \tag{7}$$

Em seguida é computada a matriz jacobiana **J** formada pelas derivadas parciais de primeira ordem do erro em relação aos parâmetros da rede, pesos sinápticos e *bias*. A matriz é mostrada na Equação 8.

$$\mathbf{J} = \begin{bmatrix} \frac{\partial e_1}{\partial w_0} & \frac{\partial e_1}{\partial w_1} & \frac{\partial e_1}{\partial w_2} & \cdots & \frac{\partial e_1}{\partial w_m} \\ \frac{\partial e_2}{\partial w_0} & \frac{\partial e_2}{\partial w_1} & \frac{\partial e_2}{\partial w_2} & \cdots & \frac{\partial e_2}{\partial w_m} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ \frac{\partial e_j}{\partial w_0} & \frac{\partial e_j}{\partial w_1} & \frac{\partial e_j}{\partial w_2} & \cdots & \frac{\partial e_j}{\partial w_m} \end{bmatrix} \tag{8}$$

Com o cálculo da matriz **J** realizado e com a definição de μ como uma constante inicial de treinamento, obtém-se o vetor de pesos da rede para a iteração $k+1$ através da Equação 9.

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (\mathbf{J}_k^T \mathbf{J}_k + \mu \mathbf{I})^{-1} \mathbf{J}_k \mathbf{e}_k \tag{9}$$

Para valores de erro total em que a condição $E_{k+1} \leq \mathrm{E}_k$ é satisfeita, a Equação 9 se aproxima da Equação 10 e o algoritmo se comporta como o algoritmo de Gauss-Newton, calculando os valores do vetor $\mathbf{w}_{k+1}$ e reduzindo μ na ordem de 10^{-1}.

$$\mathbf{w}_{k+1} = \mathbf{w}_k - (\mathbf{J}_k^T \mathbf{J}_k)^{-1} \mathbf{J}_k \mathbf{e}_k \tag{10}$$

Para iterações em que o valor de E_{k+1} é superior a E_k, o valor de μ é aumentado consideravelmente (geralmente multiplicado por 10) e o algoritmo passa a se comportar como o método do gradiente descendente, relacionando o cálculo de $\mathbf{w}_{k+1}$ com a derivada parcial de primeira ordem do erro total em relação aos parâmetros da rede como mostra a Equação 11. Ainda, nesta condição o vetor $\mathbf{w}_k$ é reiniciado e $\mathbf{w}_{k+1}$ é calculado novamente por um número determinado de vezes (usualmente 5), incluindo o aumento de μ e, então, o algoritmo volta ao seu ponto de início e realiza a próxima iteração caso o critério de parada não seja atingido.

$$\mathbf{w}_{k+1} = \mathbf{w}_k - \frac{1}{\mu} \begin{bmatrix} \frac{\partial E}{\partial w_0} & \frac{\partial E}{\partial w_1} & \frac{\partial E}{\partial w_2} & \cdots & \frac{\partial E}{\partial w_m} \end{bmatrix}^T \tag{11}$$

2.2.4 Validação

A etapa de validação avalia estatisticamente e mede o desempenho do modelo proposto. O MSE e o coeficiente de correlação podem ser empregados para medir qualitativamente o modelo que melhor estima a saída do sistema (REHMAN; MOHANDES, 2008). O cálculo do MSE dado pela Equação 12 mede a distância quadrática média entre o valor de saída estimado e o valor de saída observado, dividido pelo número de instantes de tempo do intervalo. Quanto mais próximo de 0, melhor.

$$MSE = \frac{1}{t}\sum_{i=1}^{t}(\hat{y} - y)^2 \tag{12}$$

O coeficiente R mostrado na Equação 13 mede a correlação entre as saídas estimada e observada. Valores próximos a 1 determinam relacionamento linear forte entre as funções. Já os valores próximos de 0, significam relacionamento aleatório.

$$R = \frac{\sum_{i=1}^{t}(x_i - \bar{x})^2(y_i - \bar{y})^2}{\sqrt{(\sum_{i=1}^{t}(x_i - \bar{x})^2)(\sum_{i=1}^{t}(y_i - \bar{y})^2)}} \tag{13}$$

2.3 Sinal de Excitação em SNLs

A natureza qualitativa das características de um modelo não-linear, diferentemente de casos lineares, depende de uma especificação razoavelmente completa de uma série de dados de excitação (R. K. PEARSON, 1995). Em alguns casos, segundo (AGUIRRE; S. A. BILLINGS, 1995), um sinal de entrada de frequência única pode provocar uma saída formada por sinais de frequências múltiplas e, então, além de se propor um sinal rico de excitação (amplo espectro de frequência), o perfil da amplitude do sinal também deve ser considerado no processo de identificação de sistemas não-lineares, fazendo com o que o sistema revele todas as suas características na faixa de operação de interesse.

A resposta de um SNL para um sinal de excitação equivalente a $u(t) = A_1 cos(2\pi f_1 t + \theta_1)$ pode implicar em uma resposta de uma composição de sinais de amplitudes diferentes de A_1 e frequências múltiplas de f_1, como mostrado na Equação 14 (AGUIRRE; S. A. BILLINGS, 1995).

$$y(t) = \sum_{n} B_n cos(2\pi n f_1 t + \theta_n) \tag{14}$$

Os sinais de excitação mais utilizados em SNLs são: *Maximum-Length-Sequence* (MLS) ruído branco pseudo-randômico; série de senos e cosseno de frequência variável, o qual é capaz de medir a resposta ao impulso e a distorção das harmônicas do sistema simultaneamente (FARINA, 2000).

3 METODOLOGIA

A fim de comparar as redes neuronais recorrentes contínua e discreta na modelagem de sistemas, foi feito um estudo computacional envolvendo dados simulados de um sistema de produção de petróleo por *gas-lift*. A metodologia do trabalho compreende quatro etapas: inspeção, seleção e tratamento dos dados; treinamento de diferentes modelos de rede discreta com estrutura NARX variando o número de neurônios e de atrasos na camada oculta; desenvolvimento de uma rede contínua com mesmo número de parâmetros do melhor modelo neuronal discreto; análise e comparação das redes contínua e discreta para o sistema empregado num conjunto de dados de teste. Cabe salientar que os dados empregados não possuem ruído, o que permite comparar mais diretamente os modelos.

3.1 Modelo Dinâmico Semi-Empírico

A fim de submeter as redes a um exemplo físico próximo do real, foram utilizados os dados de um experimento com um modelo semi-empírico não-linear chamado de *Fast Offshore Wells Model* (FOWM). Este modelo descreve o processo de extração de petróleo em ambiente *offshore* de águas profundas, sendo capaz de simular comportamentos não-lineares do tipo ciclo-limite, chamado de escoamento em golfadas (DIEHL et al., 2017). A descrição da forma de obtenção dos dados e o detalhamento do modelo podem ser obtidos em (FARIAS, 2018).

A base de dados fornece as séries temporais de três variáveis de entrada do sistema, isto é, a observação sequencial dos seguintes dados ao longo do tempo: a pressão do separador (P_s); a abertura da válvula Choke ($Choke$) e a vazão de *gas-lift* (G_l). As variáveis de saída disponíveis do modelo FOWM são quatro: a pressão de topo do tubo de elevação, ou *riser* (P_{rt}); a pressão na base do *riser* (P_{rb}); a pressão do topo da tubulação de injeção de gás (P_{tt}) e a pressão medida no "olho" do poço (P_{pdg}). Optou-se por utilizar esta última como variável de saída a ser modelada, já que é importante no processo e para permitir a comparação com os resultados obtidos em (FARIAS, 2018). Além disto, a escolha de três entradas e uma saída caracteriza um sistema *Multiple Input, Single Output* (MISO), simplificando a estrutura das redes a serem testadas e também a comparação entre os modelos. Os intervalos de atuação e as unidades de medida das variáveis estudadas são mostrados na Tabela 3.

Da Tabela 3, observa-se que há uma variação significativa entre as ordens de grandeza das variáveis do modelo. A fim de evitar problemas numéricos decorrentes da combinação de valores elevados com valores pequenos no modelo, a base de dados foi pré-tratada. Para tanto, os dados são normalizados a partir do escore padronizado conforme a Equação 15, em que $\mathbf{V}$ é o vetor contendo a variável de entrada ou saída. As entradas e a saída padronizadas são mostradas nas Figuras 7 e 8.

Tabela 3: Variáveis do Modelo e suas Variações

Variável	Intervalo	Unidade
G_l	$[145,000 \cdot 10^3; 205,000 \cdot 10^3]$	m^3/dia
$Choke$	[1;21]	%
P_s	$[613,250 \cdot 10^3; 1,013 \cdot 10^6]$	Pa
P_{pdg}	$[11,626 \cdot 10^6; 668,190 \cdot 10^3]$	Pa

$$s_k = \frac{v_k - média(\mathbf{V})}{DP(\mathbf{V})},\ k = 1, 2, ..., N_p \tag{15}$$

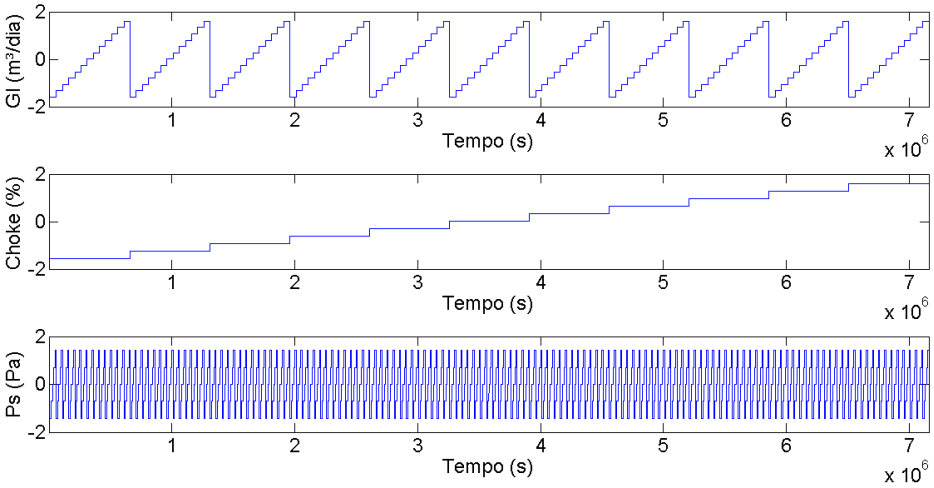

Figura 7: Entradas do Modelo FOWM
Fonte: Autor

3.1.1 Cenário de Análise

Ao se inspecionar a saída do sistema FOWM na Figura 8, observa-se que o modelo tem as características de comportamento dinâmico não-linear mais evidenciadas no final do intervalo. A resposta oscilatória sustentada correspondente ao ciclo-limite ocorre para tempos superiores a $5,8 \cdot 10^6 s$, representando o fenômeno da golfada.

Como se trata de uma base de dados de tamanho considerável, é importante do ponto de vista computacional selecionar-se adequadamente um subconjunto representativo dos dados para a modelagem caixa-preta do sistema através das RNA. Esta região deve conter informação dinâmica suficiente do efeito de variações nas três entradas para que a RNA seja treinada adequadamente. Para a fase de treinamento, portanto, foi selecionado o intervalo $[1,9 \cdot 10^6; 2,1 \cdot 10^6]$ mostrado na Figura 9 por conter variação nas três entradas do sistema.

Uma vez que uma RNA é treinada, sua capacidade de generalização deve ser testada através de uma etapa de validação, submetendo-a a condições não presentes na fase de treinamento. Esta etapa tem como objetivo validar e verificar a capacidade de predição do sistema frente à diferentes combinações das entradas. Para esta etapa é selecionado o intervalo $[2,5 \cdot 10^6; 2,7 \cdot 10^6]$ mostrado na Figura 10.

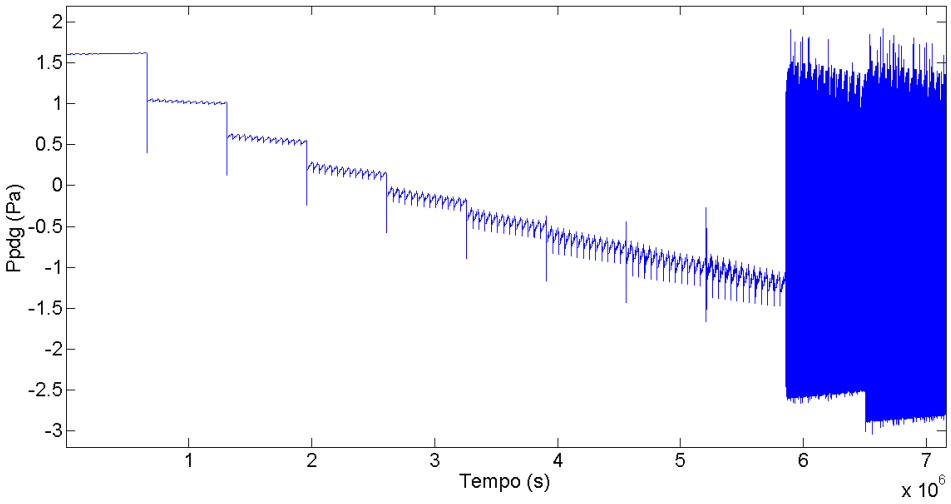

Figura 8: Saída do Modelo (P_{pdg})
Fonte: Autor

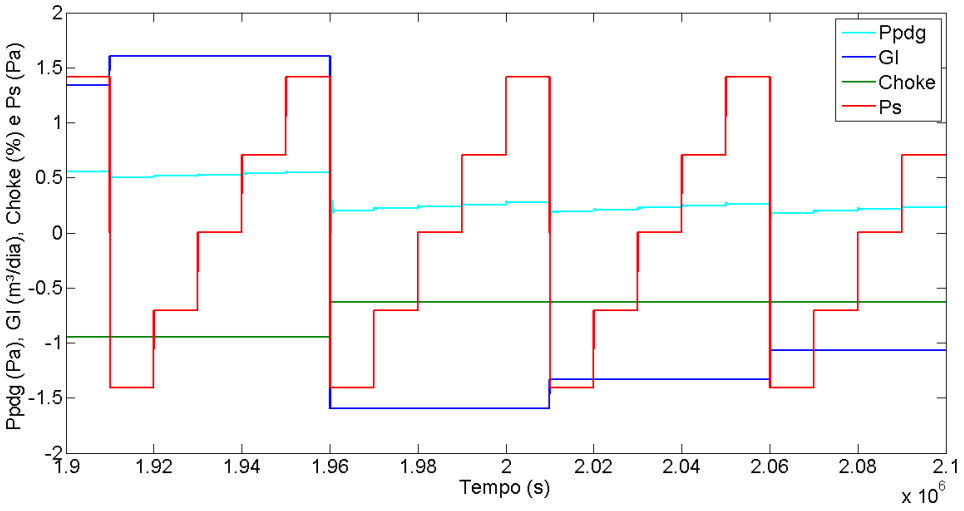

Figura 9: Intervalo de Treinamento
Fonte: Autor

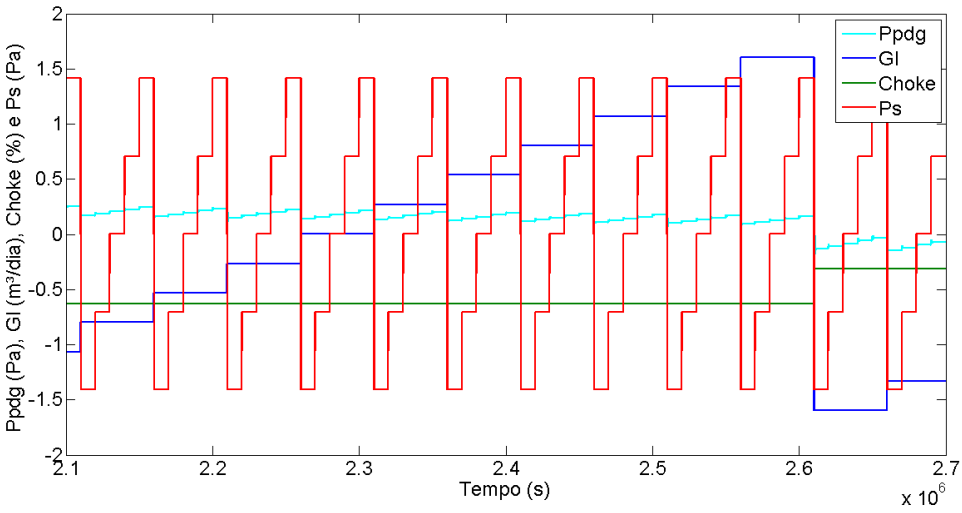

Figura 10: Intervalo de Validação
Fonte: Autor

Para comparar o desempenho das redes discreta e contínua, é importante que ambas sejam construídas sob as mesmas premissas e restrições. Portanto, as seguintes condições devem ser atendidas:

- Intervalos iguais reservados para treinamento;
- Validação no mesmo período sob mesmo estímulo de entrada;
- Arquitetura semelhante em termos do número de parâmetros, ou seja, do número de neurônios e de camadas.

Primeiramente, são gerados e validados modelos com diferentes número de neurônios ($1 \leq j \leq 10$) e atrasos ($1 \leq L \leq 3$) para que, com base nos resultados obtidos para o caso discreto, seja selecionada a arquitetura discreta de melhor desempenho e construída a rede contínua com estrutura equivalente, evitando disparidade estrutural. A seleção da arquitetura de rede mais apropriada para modelar o sistema FOWM leva em consideração os valores de MSE e de correlação *R*, conforme as definições da Seção 2.2.4 e as equações 12 e 13. Feito isto, as redes são comparadas e analisadas.

3.2 Estrutura da Rede Discreta

Neste trabalho, a estrutura NARX foi definida em termos das RNA discretas em função da sua capacidade de aprendizadem de padrões. Para tanto, foram utilizadas as ferramentas do *Neural Network Time series analysis Tool* (ntstool) do Matlab ® (R2013a). A dimensão do seu vetor de regressão $\phi(t)$ conforme a Equação 3 depende do número de atrasos e de neurônios na camada oculta como descrito na Seção 2.1. A arquitetura da rede é mostrada na Figura 11.

A rede empregada têm três camadas: de entrada, oculta e de saída. A camada de entrada é composta por um certo número de neurônios (j) a ser definido na Seção 4, assim como o número de atrasos utilizados na formação de ϕ. A função de ativação da camada oculta é a tangente hiperbólica descrita na Seção 2.2.1, uma vez que seus sinais de saída tem valores entre -1 e 1 com intercepto na origem, o que é adequado para modelar taxas

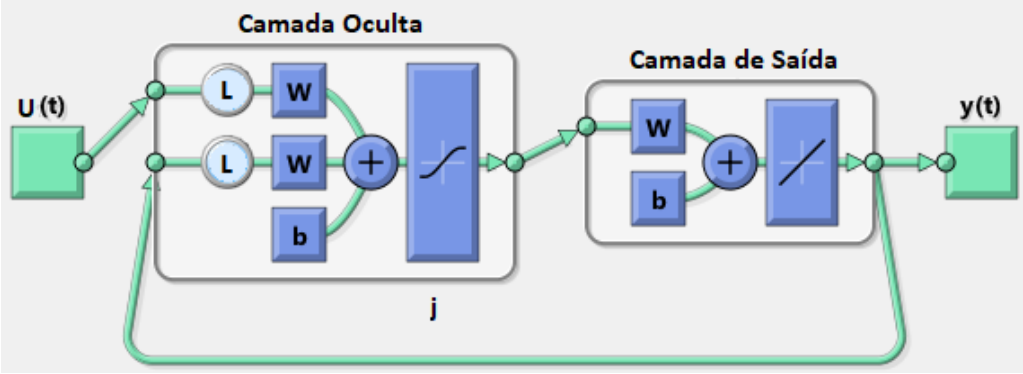

Figura 11: Arquitetura da Rede Discreta
Fonte: Adaptado do Matlab (R2013a)

de variação de sistemas físicos. Já a camada de saída é composta por pesos sinápticos (um para cada neurônio da camada oculta), um *bias* e uma função linear. É importante ressaltar que a recorrência ocorre pela conexão da saída estimada pela rede com a camada de entrada e é utilizada somente na etapa de validação, pois para a etapa de treinamento da rede (mostrada no próximo parágrafo), é utilizada a entrada real do sistema.

O treinamento e a validação da rede discreta é realizado por meio da função *discreta()* mostrada na Figura 12. Esta função está dividida em 4 partes: intervalo de treinamento; resultados do treinamento; intervalo de validação e resultados da validação. Para o intervalo de treinamento, os dados de entrada e de *target* (u e y) são tratados pela função *tonndata*, formatando-os para o padrão da interface de redes neuronais do Matlab, são definidos os números de neurônios e de atrasos na camada oculta, são preparados os dados com a função *preparets*, que preenche os estados iniciais e define a série de dados conforme o atraso selecionado, e, por fim, a rede é treinada através da função *train* pelo algoritmo de otimização de Levenberg–Marquardt descrito na Seção 2.2.3. Para o intervalo de validação, os dados são tratados e formatados novamente, porém para a entrada e o *target* do intervalo de validação, e a rede tem sua recorrência conFigurada, substituindo a saída real do sistema pela saída estimada na camada de entrada. As partes de resultados de treinamento e validação são utilizadas para calcular as saídas e as métricas de desempenho para ambas as etapas (treinamento e validação).

3.3 Estutura da Rede Contínua

O método de treinamento da rede contínua consiste em treinar a rede para o termo da taxa de variação do sistema em espaço de estados $f(x) = \frac{dy}{dt}$. Como este modelo está na forma contínua, não é usada uma representação vetorial do sistema. O código, apresentado na Figura 13, tem como objetivo minimizar a função custo apresentada na Equação 7 a partir de um chute inicial definido neste trabalho como o vetor de pesos.

Para tanto, é empregado o algoritmo de Levenberg-Marquardt (função lsqnonlin) apresentado na Seção 2.2.3 para o conjunto de dados passados como parâmetros de função (pesos, entrada, saída). A função g é obtida por meio da sub-função *AjustaPesos()*, em que são alocados como parâmetros os elementos do vetor de pesos e, então, a rede é simulada, sendo suas saídas obtidas por integração numérica, por fim retornando o valor da função custo. O final do treinamento se dá com a convergência da função objetivo para uma solução com gradiente de valor inferior a 10^{-12}, calculando um mínimo global (ou

```
function [yrd erd Rd yrd_teste erd_teste Rd_teste] = ...
    discreta(u,y,u_teste,y_teste,j,l)
%% INTERVALO DE TREINAMENTO
inputSeries = tonndata(u,false,false);
targetSeries = tonndata(y,false,false);
inputDelays = 1:l;
feedbackDelays = 1:l;
hiddenLayerSize = j;
net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize);
[inputs,inputStates,layerStates,targets] = ...
    preparets(net,inputSeries,{},targetSeries);
net.trainFcn = 'trainlm';                % LEVENBERG-MARQUARDT
net.layers{1}.transferFcn = 'tansig';    % TANGENTE-HIPERBOLICA
net.trainParam.showWindow=0;
[net,tr] = train(net,inputs,targets,inputStates,layerStates);
%% RESULTADOS DO TREINAMENTO
outputs = net(inputs,inputStates,layerStates);
yrd = fromnndata(outputs,1,false,false);
erd = sum((y(1:end-l)-yrd).^2)/length(yrd);
Cyt = corrcoef(y(1:end-l),yrd);
Rd = Cyt(2,1);
%% INTERVALO DE VALIDACAO
inputSeries = tonndata(u_teste,false,false);
targetSeries = tonndata(y_teste,false,false);
[inputs,inputStates,layerStates,targets] = ...
    preparets(net,inputSeries,{},targetSeries);
netc = closeloop(net);                   % RECORRENCIA
[xc,xic,aic,tc] = preparets(netc,inputSeries,{},targetSeries);
%% RESULTADOS DA VALIDACAO
yc = netc(xc,xic,aic);
yrd_teste = fromnndata(yc,1,false,false);
erd_teste = sum((y_teste(1:end-l)-yrd_teste).^2)/length(yrd_teste);
Cyt_teste = corrcoef(y_teste(1:end-l),yrd_teste);
Rd_teste = Cyt_teste(2,1);
end
```

Figura 12: Código em Matlab para Treinar e Validar a Rede Neuronal Discreta
Fonte: Autor

```
function pesos = TreinaRedeContinua(pesos,u,y,t)
%% PREPARACAO DOS DADOS
entrada.u1=[t,u(:,1)];
entrada.u2=[t,u(:,3)];
entrada.u3=[t,u(:,3)];
saida.y=[t,y];
target = y;
pesos = pesos';
t=t;
%% TREINAMENTO DA REDE
options = optimset('Display','iter','PlotFcns',@optimplotfval,...
    'TolFun',1e-12,'TolX',1e-12);
pesos = fminsearch(@AjustaPesos, pesos,options);
    function F = AjustaPesos(pesos)
    w0=pesos(1);
    b0=pesos(2);
    w1=pesos(3);
    w2=pesos(4);
    w3=pesos(5);
    w4=pesos(6);
    b1=pesos(7);
    RedeContinua = sim('RedeContinua_v1','SrcWorkspace', 'current');
    yrc = RedeContinua.get('yrc');
    F = sum((yrc-target).^2);
    end
end
```

Figura 13: Código em Matlab para Treinar a Rede Neuronal Contínua
Fonte: Autor

local), e com a alocação dos parâmetros obtidos na rede neuronal.

O diagrama de blocos utilizado para simular a rede contínua na chamada da subfunção *AjustaPesos()* e na função *TestaRedeContinua()* é ilustrado na Figura 13 para o caso de um único neurônio. O diagrama ilustra: a camada de entrada formada por entrada.u1 (Vazão Gl), entrada.u2 (Abertura Choke) e entrada.u3 (Ps); a camada oculta composta pelos pesos sinápticos ($w1$, $w2$, $w3$) que ponderam as entradas e a recorrência ($w4$), um *bias* para a translação do sinal e uma função de ativação do tipo tangente hiperbólica, e a camada de saída, formada por um peso sináptico, um *bias* e um integrador. É importante frisar que, assim como para a rede discreta, a recorrência da rede é utilizada apenas para o intervalo de validação, sendo utilizada a saída real do sistema para a etapa de treinamento.

Apesar de a Figura 14 conter somente 1 neurônio para fins de ilustração, é importante destacar que sua arquitetura será definida na Seção resultados, podendo haver mais neurônios e camadas.

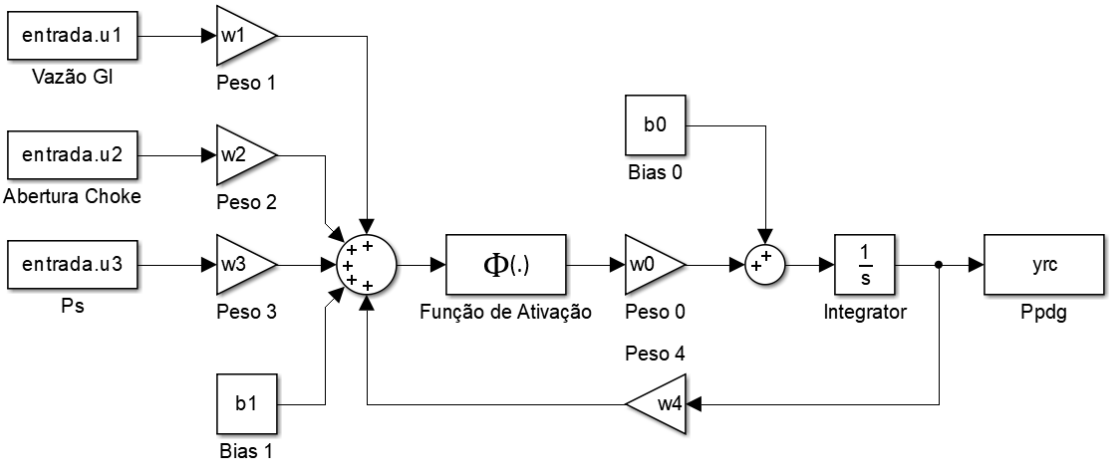

Figura 14: Diagrama de Simulação da Rede Contínua
Fonte: Autor

A rede neuronal descrita tem como função representar o comportamento da taxa de variação do estado do sistema. O estado do sistema propriamente dito é calculado no sinal de saída do bloco de integração inserido entre a retroalimentação/recorrência e o neurônio da camada de saída. O bloco yrc fornece o sinal de saída simulado.

Com os parâmetros ótimos calculados, a função *TestaRedeContinua* é executada. Ela é mostrada na Figura 15 e tem como objetivo simular o modelo otimizado e mensurar o seu desempenho para posterior análise qualitativa.

```
function [yrc erc R] = TestaRedeContinua(pesos,u,y,t)
%% PREPARACAO DOS DADOS
entrada.u1=[t,u(:,1)];
entrada.u2=[t,u(:,2)];
entrada.u3=[t,u(:,3)];
saida.y = [t,y];
target = y;
pesos = pesos';
t=t;
%% ALOCACAO DOS PESOS
w0=pesos(1);
b0=pesos(2);
w1=pesos(3);
w2=pesos(4);
w3=pesos(5);
w4=pesos(6);
b1=pesos(7);
%% SIMULACAO E RESULTADOS
RedeContinua = sim('RedeContinua_v1','SrcWorkspace', 'current');
yrc = RedeContinua.get('yrc');
erc = sum((target-yrc).^2)/length(yrc);
Cyt = corrcoef(target,yrc);
R = Cyt(2,1);
end
```

Figura 15: Código em Matlab para Testar a Rede Neuronal Contínua
Fonte: Autor

4 RESULTADOS

Nesta Seção, são apresentados os resultados das redes neuronais discreta e contínua na modelagem dos dados do sistema de produção de petróleo. Estas são analisadas em termos da estrutura e do erro de predição.

4.1 Rede Discreta

Para seleção do modelo de rede discreta, foram considerados os valores de MSE e de correlação entre a função estimada e a saída real do sistema. As tabelas Tabela 4 e Tabela 5 apresentam os valores de MSE e de R obtidos em simulação do intervalo de validação para cada modelo, totalizando 30 combinações diferentes, em que o número de neurônios é dado para cada atraso. As respostas das redes são apresentadas na forma de simulação, ou seja, sem retroalimentação dos valores reais.

Tabela 4: Valores de MSE

Nº Neurônios por Atraso	L=1	L=2	L=3
1	0,175	0,154	0,102
2	0,126	0,100	0,202
3	0,266	0,151	0,149
4	0,050	0,060	0,036
5	0,998	0,214	0,186
6	0,776	0,022	0,297
7	0,089	4,150	0,255
8	0,495	0,370	0,573
9	0,440	0,280	0,186
10	0,386	0,384	0,047

Do ponto de vista da representação em espaço de estados deste sistema, modelos com mais atrasos poderiam modelar sistemas de ordem maior. No entanto, esta afirmação não é válida para todas as variações de arquiteturas, pois, a partir da análise da Tabela 4, observa-se que arquiteturas com 7 neurônios e 2 atrasos, por exemplo, tem um disparidade significativa do erro em relação a $L = 1$ e $L = 2$. Desta forma, pode se afirmar que os valores não apresentam uma tendência clara.

Como este trabalho tem por objetivo comparar e analisar RNARs discretas e contínuas que correspondem ao modelo neuronal $f(x)$ de um sistema $f(x) = \dot{x}$, portanto, sistemas de primeira ordem, serão analisados somente arquiteturas em que $L = 1$. Da Tabela 4, tem-se que as arquiteturas de 1, 2, 4 e 7 neurônios tem erro quadrático médio inferior às

Tabela 5: Valores de R

Nº Neurônios por Atraso	L=1	L=2	L=3
1	0,962	0,963	0,967
2	0,772	0,704	-0,965
3	-0,811	0,878	-0,677
4	0,912	-0,915	0,977
5	-0,385	-0,699	-0,668
6	-0,091	0,471	-0,869
7	0,416	-0,006	-0,393
8	-0,608	-0,490	-0,463
9	-0,334	-0,924	-0,631
10	-0,788	-0,199	-0,839

demais (inferior a $0,2$), sendo candidatas para modelar o sistema. Já na Tabela 5, tem-se que as arquiteturas de 1 e 4 neurônios fornecem uma saída estimada de correlação muito alta com a saída real do sistema, eliminando as candidatas de 2 e 7 neurônios. As saídas das redes discretas de 1 e 4 neurônios são mostradas na Figura 16.

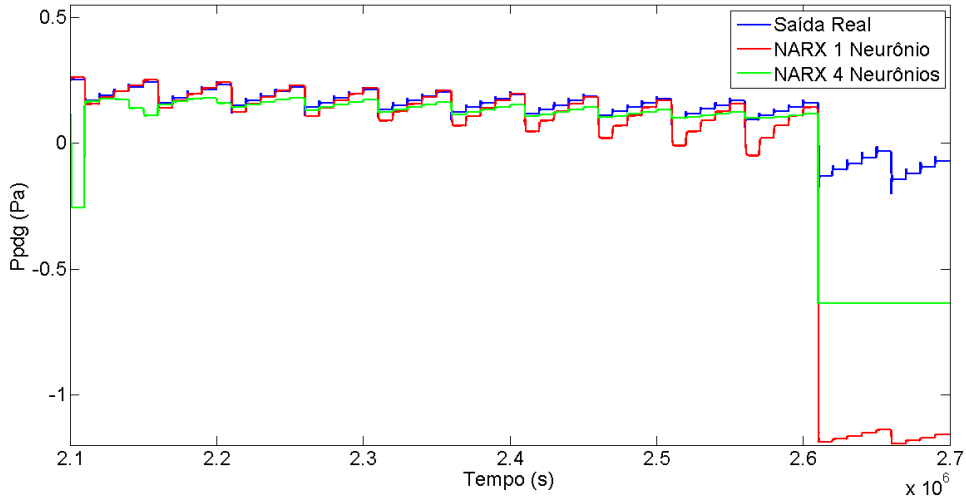

Figura 16: Saída Real e Saídas Estimadas pelas Redes Discretas de 1 e 4 Neurônios
Fonte: Autor

Da Figura 16, observa-se que a estimativa da resposta do sistema ocorre de forma adequada para as duas redes até o instante de tempo de $2,6 \cdot 10^6$. Da Seção 3.1.1 e da Figura 10, é possível notar que neste instante ocorre a variação da abertura da válvula de Choke. Apesar de haver variações dos três sinais de entrada (vazão G_l, abertura da válvula $Choke$ e pressão P_s) no intervalo de treinamento da Figura 9, a modelagem caixa-preta de sistemas não-lineares requer um sinal de excitação rico em frequência e amplitude para explorar todas as regiões do espaço de estado do sistema (AGUIRRE; S. A. BILLINGS, 1995). A rede com melhor modelagem ao efeito desta entrada é, aparentemente, a de 4 neurônios, visto que o seu MSE para a região superior ao instante de tempo $2,6 \cdot 10^6$ s é de $0,3$, enquanto que para a rede de somente 1 neurônio o erro é de $1,15$, significantemente maior. No entanto, é possível observar que o perfil de resposta obtido pela rede de 1

neurônio somente é muito semelhante ao perfil da saída real do sistema, havendo um sutil aumento de erro com o avanço do tempo e uma perda de referência para o instante de $2,6 \cdot 10^6$ s.

Tendo em vista que o intervalo de validação é 3 vezes maior que o de treinamento, pode-se esperar que, ao igualar os intervalos de treinamento e de validação, o desempenho das redes seja melhorado. Para tanto, foram calculados os valores de erro médio quadrático para o intervalo de tempo de $[2,1 \cdot 10^6]$;$2,3 \cdot 10^6]$ s, obtendo $0,189 \cdot 10^{-3}$ e $0,014$ para as redes de 1 e 4 neurônios, respectivamente. Estes resultados apontam que melhores resultados poderiam ser obtidos considerando um intervalo de dados maior para a etapa de treinamento, fazendo com que a rede aprenda mais sobre o comportamento do sistema frente a uma diversidade maior de variação de estímulos.

4.2 Rede Contínua

A fim de treinar a rede contínua para os dados gerados para o sistema FOWM e validar o modelo, é proposta inicialmente a rede de mesma arquitetura da Figura 14 com 1 neurônio somente por ser semelhante à arquitetura da rede discreta, já que a rede NARX de mesma arquitetura demonstrou resultados regulares. Após o treinamento através do algoritmo de Levenberg-Marquardt (Seção 2.2.3), são obtidos os valores de MSE e R de $0,045$ e $0,637$, respectivamente. Uma análise prévia dos valores demonstra que o valor do erro quadrático médio é relativamente baixo em comparação com os erros obtidos para os modelos discretos, porém a correlação obtida é moderada. A resposta obtida é mostrada na Figura 17.

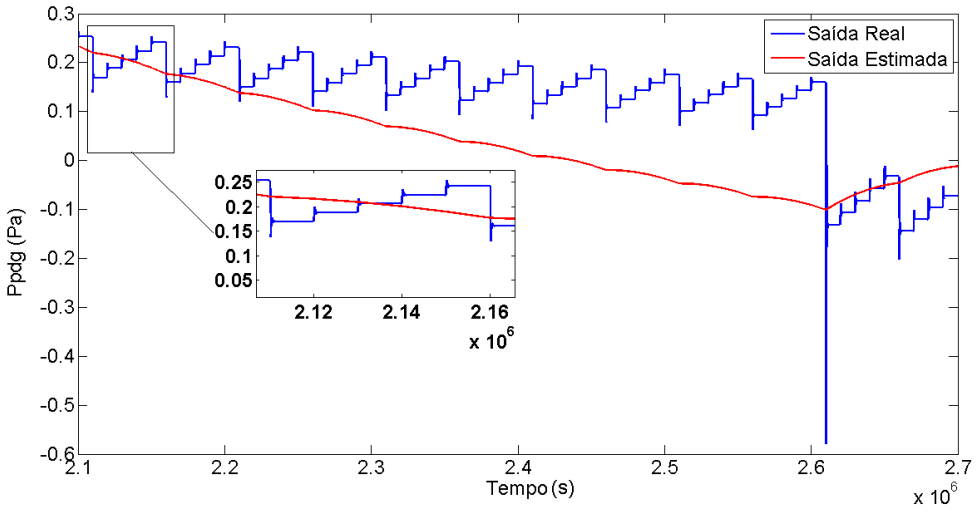

Figura 17: Saída Estimada da Rede Contínua de 1 Neurônio
Fonte: Autor

Na resposta do sistema estimada pela rede neuronal contínua na Figura 17, observa-se que a rede neuronal não consegue modelar o sistema de forma adequada. Apesar disto, o MSE após o instante $2,61 \cdot 10^6$ s é de $0,004$, ligeiramente menor que o erro obtido para o caso discreto. Além disto, este modelo deve ser evitado em regiões que ocorre o ciclo-limite causado pela variável não-periódica Gl, isto é, região que ocorre o fenômeno das golfadas a partir do instante de tempo de $2,61 \cdot 10^6$ segundos.

O resultado visto na Figura 17 demonstra resultado abaixo do esperado devido à problemas de treinamento da rede neuronal. O algoritmo utilizado para otimizar a função objetivo convergiu para um mínimo local ao invés de um mínimo global da solução, acarretando em parâmetros que levam a uma baixa capacidade de generalização da rede. Sendo assim, conclui-se que a rede contínua tem alta dependência de um ”chute”inicial razoável e de um algoritmo de treinamento de alto desempenho, capaz de fornecer uma solução global em tempo hábil, tendo em vista que para este experimento, diversos vetores de pesos e algoritmos (além do *lsqnonlin*) foram testados, como o *fminsearch*, o *Genetic Algorithm* (GA) e o *Global Search*, e também não apresentaram resultados adequados.

Com o intuito de comparar a modelagem através das RNAs contínuas e discreta, foi feita uma regressão linear sem dinâmica da saída do modelo FOWM e comparada com os resultados obtidos para as redes neuronais. A Equação 16 apresenta o modelo linear obtido para os mesmos intervalos de treinamento e validação empregado com as redes neuronais. A Figura 18 compara a resposta real do sistema com a previsão do modelo por regressão linear.

$$\hat{y} \approx -0,626 - 0,036 \cdot Gl - 1,278 \cdot Choke + 0,023 \cdot Ps \tag{16}$$

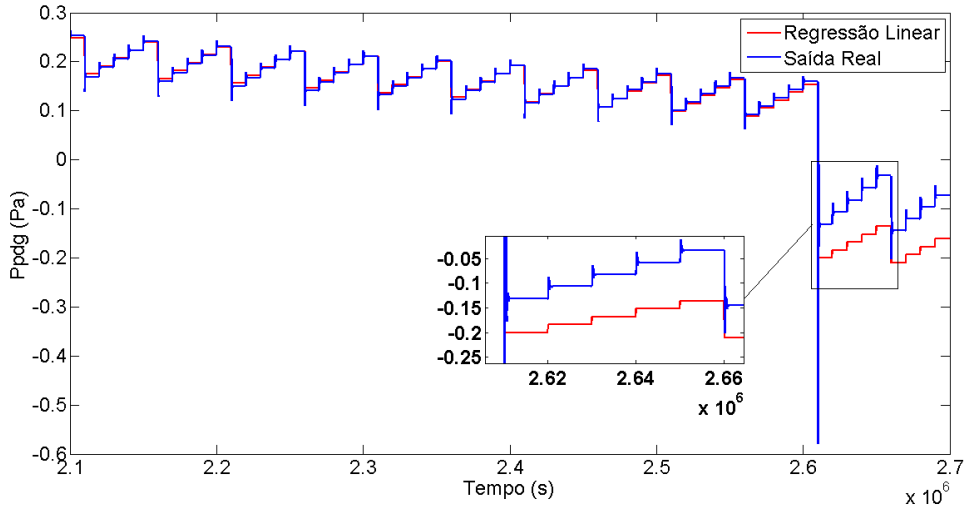

Figura 18: Comparação da Saída Real com a Previsão Obtida na Regressão Linear
Fonte: Autor

As métricas de desempenho apresentaram valores significativamente melhores que os obtidos através da modelagem neuronal: $MSE = 0,001$ e $R = 0,992$ (correlação muito alta). Ainda, da Figura 18, notavelmente se tem uma boa aproximação da saída real do sistema para tempo inferior a $2,1 \cdot 10^6$ segundos. Para tempo superior a este ponto, o erro quadrático médio obtido é de $0,007$ (menor que da rede discreta e 75% maior que da rede contínua).

A fim de analisar o desempenho de modelagem dos modelos propostos para a entrada de maior variação dentro do intervalo de treinamento, a pressão do separador (P_s), que em tese as redes neuronais teriam mais aprendizado a respeito, foram levantadas as respostas dos modelos e do sistema na Figura 19 para o intervalo.

Da Figura 19, observa-se que, apesar de o estímulo causador desta resposta no sistema ser frequente no intervalo de treinamento (cerva de 15 variações), a rede neuronal

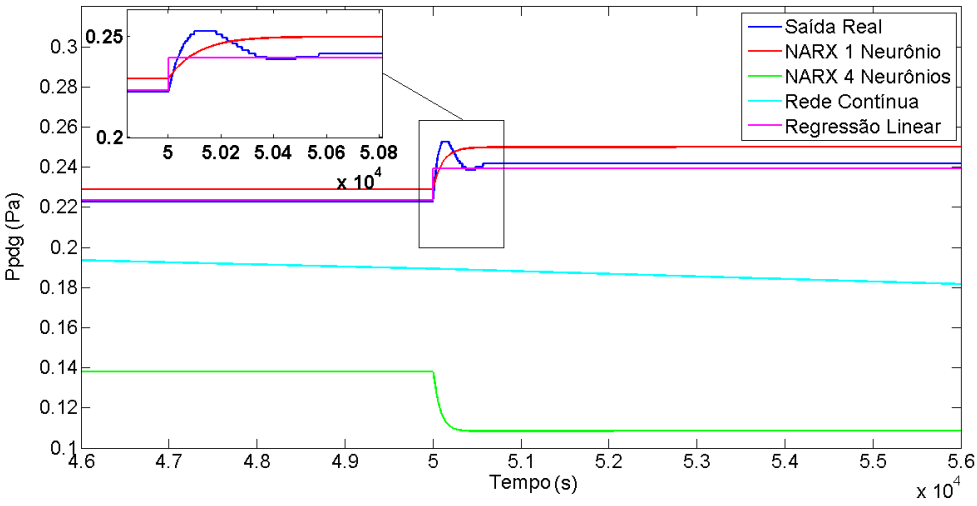

Figura 19: Comparação da Saída Real com os Modelos para um Degrau de P_s
Fonte: Autor

de 1 neurônio teve, aparentemente, a melhor aproximação da resposta transitória do sistema para o efeito desta entrada, porém não fora capaz de modelar o comportamento sub-amortecido do sistema, apresentando, uma resposta amortecida e erro em regime estacionário. Já as demais redes, a NARX de 4 neurônios e a contínua, tiveram desempenho muito inferior na modelagem da resposta do sistema para o estímulo em questão.

5 CONCLUSÃO

Neste trabalho de conclusão de curso foram apresentados e analisados modelos de rede neuronal recorrente no domínio de tempo discreto com diferentes combinações de número de neurônios e de atrasos para posterior desenvolvimento de uma rede recorrente de tempo contínuo para o termo da taxa de variação do sistema. A análise permitiu observar vantagens e desvantagens dentre os modelos estudados.

A rede contínua tem como vantagem (em comparação ao caso discreto) não depender da elaboração de um vetor de variáveis de regressão em função dos dados de entrada/saída do sistema e de atrasos da rede. No entanto, o caso contínuo possui algumas desvantagens, como a necessidade de algoritmos de otimização de alto desempenho para o cálculo assertivo (e em tempo hábil) de um mínimo global da função objetivo e a necessidade de um chute inicial adequado dos parâmetros da rede a fim de evitar comportamentos instáveis em simulação e, evidentemente, otimizar a função objetivo. Ainda, é importante destacar que o aumento de parâmetros da rede pode torná-la mais flexível para modelar sistemas de dinâmica complexa, entretanto, o cálculo de um mínimo global se torna mais complicado na medida em que o número de parâmetros aumenta, pois a dimensão espacial da função aumenta proporcionalmente.

Um aspecto fundamental a se destacar é a importância da inspeção, tratamento e definição do intervalo de dados utilizados no treinamento de uma RNAR aplicada à um sistema variante no tempo de comportamento não-linear. Considerando que o conjunto de dados utilizados para a etapa de treinamento foi de 200 mil segundos e que um conjunto 3 vezes maior foi utilizado para a etapa de validação, espera-se que, para um conjunto de dados de treinamento maior contendo mais informação a respeito da resposta do sistema frente a diferentes estímulos, a rede tenda a estimar a saída do sistema mais precisamente. No entanto, esta escolha deve ser ponderada nas premissas de projeto, uma vez que o esforço computacional necessário no cálculo dos parâmetros de um modelo neuronal aumenta proporcionalmente à base de dados fornecida ao modelo e, dependendo da aplicação, a obtenção de um conjunto maior de dados pode ser inviável devido à indisponibilidade da operação por dificuldades de implementação sensorial da aplicação.

As redes recorrentes, tanto em domínio de tempo discreto quanto em tempo contínuo, não se mostraram como opções adequadas de modelagem de um sistema dinâmico não-linear para o intervalo de treinamento e validação selecionados, havendo erro em regime estacionário, perda de referência do sinal e aumento sutil de erro quadrático médio com o incremento do tempo. No entanto, é importante frisar que um aumento do intervalo de treinamento se faz necessário, uma vez que o sistema estudado é do tipo MISO, aumentando a dificuldade de mapeamento das respostas do sistema.

A técnica de regressão linear tem se mostrado adequada frente às redes neuronais testadas, visto que seu MSE para o intervalo de validação é aproximadamente 4 vezes

menor que o erro da rede neuronal de melhor desempenho para todo o intervalo. Também, como esta técnica possui pouco esforço computacional em comparação com os métodos de treinamento neuronais, ela pode ser um modelo adequado para representar o sistema (em relação às redes recorrentes estudadas neste trabalho).

A modelagem de um sistema dinâmico não-linear tem se mostrado não trivial. Outras abordagens podem ser tomadas em trabalhos futuros. Ao invés de restringir o treinamento do modelo neuronal para o termo da taxa de variação do sistema, a abrangência de mais estados do sistema pode ser uma oportunidade de melhoria, visto que uma gama maior de comportamentos e fenômenos podem ser modelados.

Para trabalhos futuros, sugere-se o estudo de redes neuronais recorrentes que mapeiem mais estados do sistema, inserindo atrasos no modelo discreto e blocos de integração numérica para o caso contínua. Ainda, sugere-se a implementação de outras arquiteturas, como a rede de Hopfield, NARMAX, Filtro de Kalman Estendido (FKE), árvores de decisão, entre outros.

BIBLIOGRAFIA

AGUIRRE, L. A.; BILLINGS, S. A. Dynamical effects of overparametrization in nonlinear models. **Physica D: Nonlinear Phenomena**, Elsevier, v. 80, n. 1-2, p. 26–40, 1995.

BENGIO, Y. et al. Recurrent neural networks for adaptive temporal processing. **Universitá di Firenze**, 1993.

CHEN, S.; BILLINGS, S. Neural networks for nonlinear dynamic system modelling and identification. **International journal of control**, Taylor & Francis, v. 56, n. 2, p. 319–346, 1992.

CHEN, S.; BILLINGS, S.; GRANT, P. Non-linear system identification using neural networks. **International journal of control**, Taylor & Francis, v. 51, n. 6, p. 1191–1214, 1990.

CHOW, T. W.; LI, X.-D. Modeling of continuous time dynamical systems with input by recurrent neural networks. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, IEEE, v. 47, n. 4, p. 575–578, 2000.

CHURCHLAND, P.; SEJNOWSKI, T. **The Computational Brain. Bradford Book**. [S.l.]: MIT Press Cambridge: 1992.

CS231N. **Convolutional Neural Networks for Visual Recognition**. [S.l.: s.n.], 2019. [Online; accessado em 20-Abril-2019]. Disponível em: ¡`http://cs231n.github.io/neural-networks-1/`¿.

DEKA, A. et al. Predictive modeling techniques to forecast energy demand in the United States: A focus on economic and demographic factors. **Journal of Energy Resources Technology**, American Society of Mechanical Engineers, v. 138, n. 2, p. 022001, 2016.

DI MEGLIO, F. et al. Stabilization of slugging in oil production facilities with or without upstream pressure sensors. **Journal of Process Control**, Elsevier, v. 22, n. 4, p. 809–822, 2012.

DIEHL, F. C. et al. Fast Offshore Wells Model (FOWM): A practical dynamic model for multiphase oil production systems in deepwater and ultra-deepwater scenarios. **Computers & Chemical Engineering**, Elsevier, v. 99, p. 304–313, 2017.

FARIAS, N. F. Desenvolvimento de analisador virtual para predição da pressão de fundo em poços de petróleo utilizando rede neural, 2018.

FARINA, A. Simultaneous measurement of impulse response and distortion with a swept-sine technique. In: AUDIO ENGINEERING SOCIETY. AUDIO Engineering Society Convention 108. [S.l.: s.n.], 2000.

FUNAHASHI, K.-i.; NAKAMURA, Y. Approximation of dynamical systems by continuous time recurrent neural networks. **Neural networks**, Elsevier, v. 6, n. 6, p. 801–806, 1993.

GÉRON, A. **Hands-on machine learning with Scikit-Learn and TensorFlow: concepts, tools, and techniques to build intelligent systems**. [S.l.]: "O'Reilly Media, Inc.", 2017.

HAYKIN, S. **Redes neurais: princıpios e prática**. [S.l.]: Bookman Editora, 2007.

HOPFIELD, J. J. Neural networks and physical systems with emergent collective computational abilities. **Proceedings of the national academy of sciences**, National Acad Sciences, v. 79, n. 8, p. 2554–2558, 1982.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural networks**, Elsevier, v. 2, n. 5, p. 359–366, 1989.

LEONTARITIS, I.; BILLINGS, S. A. Input-output parametric models for non-linear systems part I: deterministic non-linear systems. **International journal of control**, Taylor & Francis, v. 41, n. 2, p. 303–328, 1985.

LEVENBERG, K. A method for the solution of certain non-linear problems in least squares. **Quarterly of applied mathematics**, v. 2, n. 2, p. 164–168, 1944.

LJUNG, L. System identification. **Wiley Encyclopedia of Electrical and Electronics Engineering**, Wiley Online Library, p. 1–19, 1999.

LJUNG, L.; SÖDERSTRÖM, T. **Theory and practice of recursive identification**. [S.l.]: MIT press, 1983.

MARQUARDT, D. W. An algorithm for least-squares estimation of nonlinear parameters. **Journal of the society for Industrial and Applied Mathematics**, SIAM, v. 11, n. 2, p. 431–441, 1963.

MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, 1943.

OGATA, K. Modern control engineering. **Book Reviews**, v. 35, n. 1181, p. 1184, 1999.

PÁDUA BRAGA, A. de. **Redes neurais artificiais: teoria e aplicações**. [S.l.]: LTC Editora, 2000. ISBN 9788521615644. Disponível em: ¡`https://books.google.com.br/books?id=R-p1GwAACAAJ`¿.

PEARSON, R. Selecting nonlinear model structures for computer control. **Journal of process control**, Elsevier, v. 13, n. 1, p. 1–26, 2003.

PEARSON, R. K. Nonlinear input/output modelling. **Journal of Process Control**, Elsevier, v. 5, n. 4, p. 197–211, 1995.

PINDYCK, R. S.; RUBINFELD, D. L. **Econometric models and economic forecasts**. [S.l.]: Irwin/McGraw-Hill Boston, 1998. v. 4.

REHMAN, S.; MOHANDES, M. Artificial neural network estimation of global solar radiation using air temperature and relative humidity. **Energy Policy**, Elsevier, v. 36, n. 2, p. 571–576, 2008.

SCHMIDT, Z.; BRILL, J.; BEGGS, H. Choking can eliminate severe pipeline slugging. **Oil & gas journal**, PENNWELL PUBL CO ENERGY GROUP 1421 S SHERIDAN RD PO BOX 1260, TULSA, OK 74101, v. 77, n. 46, p. 230, 1979.

SHEPHERD, G. M. **The synaptic organization of the brain**. [S.l.]: Oxford university press, 2003.

SJÖBERG, J. et al. Nonlinear black-box modeling in system identification: a unified overview. **Automatica**, Elsevier, v. 31, n. 12, p. 1691–1724, 1995.

UNBEHAUEN, H.; RAO, G. Continuous-time approaches to system identification—a survey. **Automatica**, Elsevier, v. 26, n. 1, p. 23–35, 1990.

YU, H.; WILAMOWSKI, B. M. Levenberg-marquardt training. **Industrial electronics handbook**, CRC Press Boca Raton, FL, v. 5, n. 12, p. 1, 2011.

ANEXO A CÓDIGOS EM PYTHON

A.1 Código Utilizado para Extrair e Estruturar Dados de Ensaio

```
Created on Sun Oct  7 16:13:26 2018
@author: Milton Farias

# IMPORTA AS FUNCOES
import pickle
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
import matplotlib as plt

# CAMINHO DAS BASES
import pandas as pd
pd.set_option('display.max_columns', 500)
base1=pd.read_csv(r'C:\bases\saidas.csv',sep=',',decimal='.',...
    header=None)
base1=base1.rename(columns={0:'Prt',1:'Prb',2:'Ptt',3:'Ppdg'})
base1.columns

# APAGA COLUNAS SOBRESSALENTES
lista=[i for i in range(4,11)]
base1=base1.drop(lista,1)

# BASE 2
base2=pd.read_csv(r'C:\bases\entradas.csv',sep=',',decimal='.',...
    header=None)

# CABECALHO
base2=base2.rename(columns={0:'Gl',1:'Choke',2:'Ps'})
base2.columns

# CONCATENA BASES
base1 = pd.concat([base2,base1],1)
base1.head()
base1 = base1.reset_index()
base1=base1.rename(columns={'index':'contador'})
```