



UNIVERSIDADE FEDERAL DA PARAÍBA
CENTRO DE ENERGIAS ALTERNATIVAS E RENOVÁVEIS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

Jefferson Whitney de Miranda Mendonça

Desenvolvimento de um Modelo de Previsão de Demanda de Energia Baseado em Algoritmos de Inteligência Artificial de Curto Prazo

João Pessoa - PB

2022

Jefferson Whitney de Miranda Mendonça

Desenvolvimento de um Modelo de Previsão de Demanda de Energia Baseado em Algoritmos de Inteligência Artificial de Curto Prazo

Trabalho de Conclusão de Curso apresentado
à Universidade Federal da Paraíba como exi-
gência para a obtenção do título de Bacharel
em 2022.

Universidade Federal da Paraíba

Centro de Energias Alternativas e Renováveis

Curso de Graduação em Engenharia Elétrica

Orientador: Prof. Dr. Juan Mauricio Villanueva

João Pessoa - PB

2022

© Jefferson Whitney de Miranda Mendonça

Catálogo na publicação
Seção de Catalogação e Classificação

M539d Mendonca, Jefferson Whitney de Miranda.
Desenvolvimento de um Modelo de Previsão de Demanda
de Energia Baseado em Algoritmos de Inteligência
Artificial de Curto Prazo / Jefferson Whitney de
Miranda Mendonca. - João Pessoa, 2022.
53 f. : il.

Orientação: Prof Dr Juan Mauricio Villanueva.
TCC (Graduação) - UFPB/CEAR.

1. Previsão de Demanda Energética. 2. Distribuição
de Energia Elétrica. 3. Previsão de Séries Semporais.
4. Redes Neurais Artificiais. 5. Inteligência
Artificial. I. Villanueva, Prof Dr Juan Mauricio. II.
Título.

UFPB/CT

CDU 621.317.38(043.2)

Jefferson Whitney de Miranda Mendonça

Desenvolvimento de um Modelo de Previsão de Demanda de Energia Baseado em Algoritmos de Inteligência Artificial de Curto Prazo

Trabalho de Conclusão de Curso apresentado à Universidade Federal da Paraíba como exigência para a obtenção do título de Bacharel em 2022.

Aprovado em: 07/12/2022

Nota: 10,0


Prof. Dr. Juan Mauricio Villanueva
Orientador


Prof. Dr. Alexandre José Virginio dos Santos
Examinador Interno UFPB


Prof. Dr. Cicero Da Rocha Souto
Examinador Interno UFPB

João Pessoa - PB
2022

Agradecimentos

Nessas palavras, agradeço imensamente a todos que, de alguma forma, me ajudaram direta e indiretamente a chegar a este momento de conclusão da minha graduação.

Agradeço primeiramente aos meus pais, Juarez Carlos e Kid Cássia que com muito carinho, amor, esforço, dedicação e suor, conseguiram me apoiar durante todo o meu curso, para que eu pudesse me preocupar apenas em estudar.

Agradeço ao meu irmão Jefte Kerison, que me ajudou a rir nos momentos mais difíceis e me escutou quando mais eu precisava.

Agradeço aos meus avós e tio, Djalma Salustiano, Josefa Adelaide e Kleber Mendonça que me apoiaram e me deram carinho de todas as formas para que eu pudesse continuar.

Aos amigos que construí durante todo o curso, todas as turmas que participei, minha eterna gratidão por toda a parceria durante esses anos de curso, pelas noites de estudos na UFPB, por todas as provas em conjunto durante a pandemia do COVID-19, pelo apoio em diversos momentos da vida acadêmica, pessoal e profissional.

Em especial ao grupo de pesquisa UFPBaja que foram, nos meus primeiros degraus da minha vida profissional, de extrema importância no meu crescimento, tanto como pessoa como um engenheiro.

Aos professores que corrigiram as provas com carinho e tiveram empatia. A aqueles que me ajudaram e me guiaram em projetos de pesquisa, são um dos pilares que me ajudaram a me sustentar no mundo corporativo.

Aos meus amigos que criei em Belo Jardim - PE, que me apoiaram e me ajudaram no meu desenvolvimento pessoal e profissional.

Ao professor Juan que me acompanhou durante grande parte do curso e que me ajudou a terminar essa última etapa de todo esse processo árduo, moroso mas gratificante.

*“Então peguemos de volta o que nos foi tirado, mano,
ou você faz isso ou seria em vão o que os
nossos ancestrais teriam sangrado”
(Djonga, 2019)*

Resumo

As pesquisas e estudos com o objetivo de previsão de demanda de energia elétrica são de extrema importância para concessionárias de energia elétrica em todo o mundo, pois quando se é obtido uma boa previsão de consumo é possível executar um bom planejamento de médio e longo prazo, evitando sobre-geração e sub-geração de energia elétrica. As empresas que fornecem energia elétrica precisam de softwares e recursos computacionais para estimar suas previsões. Neste trabalho, utilizando softwares e IDE's como Visual Studio Code® e Google Colab® juntamente com bibliotecas voltadas a redes neurais artificiais foram realizadas estimativas de previsões de demanda de energia utilizando dados de uma subestação de energia elétrica. Para realizar uma previsão mais próxima possível da curva real, neste projeto foram utilizadas técnicas de LSTM em conjunto com a normalização dos dados de entrada, parâmetro de validação da própria rede neural e métodos iterativos para encontrar a melhor configuração de quantidade de neurônios em camadas recursivas diferentes. Os dados de potência fornecida diariamente foram obtidos de uma subestação de energia elétrica obtidas por um sistema SCADA no período de 2008 a 2013, que foram tratados para o uso neste trabalho.

Palavras-chave: Redes Neurais Artificiais. Previsão de Demanda de Energia Elétrica. LSTM . *Dropout*.

Abstract

Research and studies with the objective of forecasting electricity consumption are extremely important for electric utilities around the world, because when a good forecast of consumption is obtained, it is possible to carry out good medium and long-term planning, avoiding over-generation and under-generation of electricity. Companies that supply electricity, need software and computational resources to estimate their forecasts. In this study, using software and IDE's such as Visual Studio Code® and Google Colab® together with specific libraries with neural networks, estimates of energy demand forecast were created using data from an electrical power substation. In order to perform a prediction as close as possible to the real curve, in this project, LSTM techniques were used together with the normalization of the input data, validation parameter of the neural network itself and iterative methods to find the best configuration of number of neurons in many different recursive layers. The daily supplied power data were obtained from an electric power substation obtained by a SCADA system in the period from 2008 to 2013, which were treated for use in this work.

Keywords: Artificial Neural Network. Electrical Energy Consumption Forecast. LSTM. Dropout.

Lista de ilustrações

Figura 1 – Representação de um sistema elétrico de potência.	19
Figura 2 – Sistema de aquisição de dados.	22
Figura 3 – Arquitetura básica da rede neural recorrente.	23
Figura 4 – Arquitetura de rede de memória de longo prazo.	24
Figura 5 – Plot da função sigmoide.	24
Figura 6 – Primeira parte da LSTM	25
Figura 7 – Segunda parte da LSTM	26
Figura 8 – Plot da função Tangente Hiperbólica	26
Figura 9 – Terceira parte da LSTM	27
Figura 10 – Quarta parte da LSTM	28
Figura 11 – Rede neural padrão sem o <i>Dropout</i>	30
Figura 12 – Rede neural padrão	31
Figura 13 – Camada em treino com a probabilidade p e saída com pesos W	32
Figura 14 – Na etapa de teste, a camada com os pesos sendo multiplicados por p	33
Figura 15 – Diagrama unifilar	34
Figura 16 – Esquema dos elementos básicos de um monitoramento de subestações e sistema SCADA	35
Figura 17 – Série temporal de demanda para o período de 2008-2013	36
Figura 18 – Modelo de entradas e saídas da RNA	37
Figura 19 – Modelo Sequencial	38
Figura 20 – Modelo LSTM	39
Figura 21 – Bibliotecas	40
Figura 22 – Comparação de curvas das métricas com e sem validação.	43
Figura 23 – Variações de <i>Dropout</i> com normalização	45
Figura 24 – Variações de <i>Dropout</i> sem normalização	46
Figura 25 – Melhor configuração de neurônios na LSTM com <i>Dropout</i> 0.3.	47
Figura 26 – Melhor configuração de neurônios na LSTM	48
Figura 27 – Gráfico de previsões usando $dp=0$	49
Figura 28 – Gráfico de previsões usando $dp=0.3$	49

Lista de tabelas

Tabela 1 – Métricas da LSTM com validação e sem validação.	42
Tabela 2 – Métricas da LSTM com normalização e com seus respectivos valores de <i>Dropout</i>	44
Tabela 3 – Métricas da LSTM sem normalização e seus respectivos valores de <i>Dropout</i>	45
Tabela 4 – Métricas da melhor configuração da LSTM com <i>Dropout</i> 0.3.	47
Tabela 5 – Métricas da melhor configuração da LSTM com <i>Dropout</i> 0.	48

Lista de abreviaturas e siglas

AR	Auto Regressivo
DNNs	Deep Neural Networks
dp	<i>Dropout</i>
MSD	Desvio Médio Quadrático
EPE	Empresa de Pesquisa Energética
MSE	Erro Médio Quadrático
MAPE	Erro Percentual Médio Absoluto
MLP	Feedforward Multilayer Perceptron
ReLU	Função de ativação linear retificada
LSTM	Long-Short-Term-Memory
MA	Média Móvel
ARIMA	Média Móvel Integrada
GARCH	Modelo de Heterocedasticidade Condicional Autorregressiva Generalizada
PIB	Produto Interno Bruto
RAM	Random Access Memories
RNAs	Rede Neural Artificial
RTU	Unidade Terminal Remota
GRNN	Rede Neural de Regressão Generalizada
SVR	Regressão Vetorial de Suporte
SEP	Sistema Elétrico de Potência
SIN	Sistema Interligado Nacional
SCADA	Supervisory Control and Data Acquisition
tanh	Tangente hiperbólica

Lista de símbolos

α	Letra grega Alpha
σ	Letra grega Sigma
Σ	Letra grega maiúscula Sigma

Sumário

1	INTRODUÇÃO	14
1.1	Considerações Preliminares	14
1.2	Motivação	16
1.3	Objetivos	17
1.3.1	Objetivo Geral	17
1.3.2	Objetivos Específicos	17
1.4	Organização do Trabalho	17
2	FUNDAMENTAÇÃO TEÓRICA	19
2.1	Sistema Elétrico de Potência	19
2.1.1	Subsistema: Geração	20
2.1.2	Subsistema: Transmissão	20
2.1.3	Subsistema: Distribuição	20
2.2	LSTM	22
2.2.1	Estrutura interna da LSTM	23
2.2.2	Passo a passo da LSTM	25
2.3	Métricas de avaliação	28
2.3.1	Erro médio quadrático (MSE)	29
2.3.2	Mean absolute percentage error (MAPE)	29
2.3.3	Erro máximo	30
2.4	Dropout	30
3	METODOLOGIA PROPOSTA	34
3.1	Apresentação do estudo de caso	34
3.1.1	Modelagem do Banco de dados	36
3.1.2	Previsão de curto prazo de máxima semanal usando LSTM	37
3.1.3	Metodologia	40
4	RESULTADOS	42
4.1	Utilização do recurso "Validação" da LSTM	42
4.2	Normalização dos dados de entrada	43
4.2.1	Variações de <i>Dropout</i>	43
4.2.1.1	Variações de <i>Dropout</i> com normalização dos dados de entrada	44
4.2.1.2	Variações de <i>Dropout</i> sem normalização dos dados de entrada	45
4.3	Teste iterativo de nós para LSTM	46

5	CONCLUSÃO	47
6	TRABALHOS FUTUROS	50
	REFERÊNCIAS	51

1 Introdução

1.1 Considerações Preliminares

Com o surgimento da China como o maior consumidor de energia do mundo, a conscientização sobre o consumo de energia dos países em desenvolvimento aumentou. De acordo com cenários econômicos comuns, o resto do mundo em desenvolvimento provavelmente também verá uma expansão econômica. Com este crescimento, certamente virá o crescimento rápido e contínuo da demanda de energia. Em 2000, apenas 66% dos lares do mundo em desenvolvimento tinham acesso à eletricidade. As taxas de aquisição de eletrodomésticos permanecem baixas, mas com melhor acesso à eletricidade e uma renda mais alta, pode-se esperar que as famílias vejam seu consumo de energia elétrica aumentar significativamente (LETSCHERT, 2008).

Assim como em outros países em desenvolvimento, a demanda por energia também está aumentando no Brasil. Por exemplo, entre 1990 e 2009, a taxa média de crescimento do consumo de energia no país foi de 3,1% ao ano, enquanto o volume do PIB cresceu em média 3,9% ao ano no mesmo período. O consumo de energia per capita variou de 1.85 kWh em 1990 a 2.225 kWh em 2009. Para acompanhar um desenvolvimento sustentável, é preciso que uma determinada região possua um planejamento de como o fornecimento desses recursos será feito no longo prazo para que não haja excessos ou faltas (NETO; CORRÊA; PEROBELLI, 2016).

O planejamento de longo prazo da oferta-demanda de energia deve satisfazer as exigências do desenvolvimento sustentável dos países. Previsões precisas podem ajudar os tomadores de decisão a conhecer o volume e a tendência do consumo futuro de energia para melhor programar e planejar as operações do sistema de abastecimento (YAO, 2013).

Com o aumento da dependência energética vem a alta demanda, e tendo em vista que, como é difícil armazenar alguns tipos de energia, principalmente a eletricidade, é mais importante distribuir a energia com o menor custo e desperdício. Para isso existe o planejamento eficiente da distribuição de energia que precisa de previsões precisas da demanda futura para fazer o equilíbrio entre a oferta e a demanda de energia (AN et al., 2013).

Durante as previsões e o planejamento, a subestimação do consumo de energia pode levar à falta de energia, o que pode ser prejudicial tanto para a economia quanto para o cotidiano da sociedade. Por outro lado, a superestimação da demanda de energia pode levar à criação de capacidade não utilizada que equivale ao desperdício de recursos, principalmente financeiros. Portanto, usar modelos para prever com precisão as tendências

futuras de consumo de energia especificamente com dados não lineares é uma questão importante para os sistemas de produção e distribuição de energia (KAVAKLIOGLU et al., 2009).

A previsão de demanda energética desempenha um papel vital nas redes de distribuição. A previsão de demanda energética de curto prazo prevê a carga que é de várias horas a várias semanas à frente. E por não ser linear, a natureza não estacionária e não sazonal da demanda de energia elétrica de curto prazo, a torna um processo desafiador de previsão (ZHENG et al., 2017).

Como a demanda energética, é, principalmente, uma série temporal univariada (ALMESHAI EI; SOLTAN, 2011), muitos métodos gerais de previsão de séries temporais podem ser aplicados para previsão de carga elétrica. Uma classe geral de métodos são os modelos estatísticos de previsão que incluem os Auto Regressivos (AR), Média Móvel (MA), Modelos Autorregressivos de Média Móvel Integrada (ARIMA) (ZHANG, 2003) (KHASHEI; BIJARI, 2011) e inúmeras variantes (CHEN; WANG, 2007). Em particular, o ARIMA que é um dos métodos mais populares e comumente utilizados para previsão de séries temporais. No entanto, esses métodos funcionam sob a suposição de que a série temporal observada e a série temporal futura são linearmente relacionadas, o que as torna menos eficazes para séries temporais com características não lineares. Há muitos trabalhos que explicam as suas aplicações em previsões não lineares, como vemos a GARCH, Modelo de heterocedasticidade condicional autorregressiva generalizada (HOR; WATSON; MAJITHIA, 2006).

No entanto, estes modelos estatísticos só têm um bom desempenho de previsão ao longo de dados estacionários (KAM, 2014) (LINEESH; MINU; JOHN, 2010), enquanto a série temporal de demanda energética, na maioria das vezes, pode não ser estacionária.

Outra classe de métodos de previsão é baseada em Redes Neurais Artificiais (RNAs). As RNAs tornaram-se imensamente populares na previsão de demanda energética na última década. Basicamente, as RNAs imitam o cérebro humano para aprender regularidades e padrões automaticamente da experiência passada e produzem resultados generalizados. Em contraste com o sistema linear baseado em ARIMA, as RNAs são um conjunto de métodos que são orientados por dados, o que significa que não há necessidade de qualquer conhecimento prévio da relação entre os modelos de previsão e os dados. As RNAs geralmente obtêm bons resultados, especialmente para modelos e séries temporais (PANIGRAHI; KARALI; BEHERA, 2013). Existe uma extensa literatura sobre o uso de RNAs para previsão de carga elétrica, como *Feedforward Multilayer Perceptron* (MLP) (GARCÍA-ASCANIO; MATÉ, 2010), Rede Neural de Regressão Generalizada (GRNN) (LI; WANG, 2012), Regressão vetorial de suporte (SVR) (KO; LEE, 2013), etc.

Embora exista uma extensa pesquisa sobre o tema, a previsão precisa que os dados

de demanda energética sejam contínuos, caso contrário acaba sendo um desafio para redes inteligentes. A previsão de demanda energética, é, geralmente, um problema de previsão de séries temporais univariadas, que é mais desafiador do que o problema de previsão de séries temporais multivariadas correspondente. O motivo dessa diferença é porque não há informações adicionais de outros dados fontes que podem ser utilizadas para a aprendizagem (SAK; SENIOR; BEAUFAYS, 2014). Além disso, se analisarmos séries temporais lineares, estacionárias e sazonais, vemos que as séries temporais de demanda energética em um sistema de energia de pequena escala são não lineares, não estacionários e não sazonais, onde não sazonal significa a sua não periodicidade aparente no tempo. Isso torna difícil a previsão com precisão para tais séries temporais em um longo horizonte de tempo.

Portanto, mais esforços são necessários para desenvolver métodos de previsão. Neste trabalho, abordaremos esse desafio utilizando o *Long-Short-Term-Memory* (LSTM) (HOCHREITER; SCHMIDHUBER, 1997), que é um recurso especial das redes neurais recorrentes, que pode ser utilizado para aprender mais sequências temporais e dependências de longo prazo com a mesma precisão do que uma *Deep Neural Networks* (DNNs) ou uma RNNs convencional (SAK; SENIOR; BEAUFAYS, 2014). Será desenvolvido um novo esquema de previsão de demanda energética baseado em LSTM, no qual esta configuração será capaz de prever com precisão os dados que até então eram complexos, sendo não lineares e não estacionários, abrindo a possibilidade para analisar até em períodos não sazonais ao longo de um longo horizonte de previsão, que também será otimizado utilizando de técnicas e filtros em métricas avaliativas, para criar o melhor modelo possível.

1.2 Motivação

A crescente utilização de energia elétrica no Brasil causa um impacto no nosso sistema elétrico de distribuição de energia, exigindo um estudo do aumento da demanda e os impactos que esse consumo gera no desempenho da nossa rede elétrica. Para acompanhar esse crescimento é necessário um planejamento de geração com objetivo geral de diminuir os custos de sua produção.

Um dos principais problemas acerca da geração de energia elétrica é a não previsibilidade do consumo, podendo ocasionar déficit ou geração excedente de energia, que pode levar ao desperdício de energia produzida em excesso ou a falta de energia não planejada que deveria ser distribuída. A variação de consumo ocorre devido a fatores diversos durante todo o ano, sendo épocas de festas e feriados ou devido a sazonalidade periódica anual, apresentando altas e baixas demandas durante o ano que podem ser evitadas com seu devido planejamento futuro.

Com base neste pressuposto, este trabalho abordará os desafios técnicos da previsão

de demanda energética e adicionalmente, será apresentada uma proposta para minimizar o problema de previsão de demanda nas redes de distribuição que tenham a variação de consumo variante, através do desenvolvimento de uma rede neural inteligente visando a obtenção de valores de consumo energético futuro a fim de serem supridas pela rede fornecedora, de forma a reduzir a sobretensão e sobtensão nas redes elétricas de distribuição utilizando de métricas avaliativas e técnicas de redes neurais avançadas.

1.3 Objetivos

1.3.1 Objetivo Geral

O objetivo deste trabalho é apresentar e trabalhar em um histórico de demanda energética de uma empresa privada, com um enfoque especial na potência fornecida, e propor um método de previsão da demanda de energia elétrica requisitada, que poderá servir de suporte na tomada de decisão para planejamento de produção de energia elétrica.

1.3.2 Objetivos Específicos

Como objetivos específicos tem-se:

- Introduzir conceitos básicos da previsão da demanda de energia elétrica
- Introduzir a aplicação do conceito de *Machine Learning* na previsão demanda de energia elétrica
- Apresentar a modelagem utilizada para a previsão da demanda de energia elétrica
- Utilizar de ferramentas para a melhoria de desempenho da rede neural, objetivando a comprovação da eficiência de suas previsões
- Utilizar do método iterativo para obter a melhor configuração da rede neural utilizada
- Propor discussões a partir das informações e resultados obtidos.

1.4 Organização do Trabalho

O trabalho, além desta introdução, organiza-se em 6 capítulos da seguinte forma:

- Capítulo 1 - No primeiro capítulo é apresentada uma introdução sobre o trabalho, contando com a motivação e os objetivos.

-
- Capítulo 2 - No segundo capítulo é feita uma fundamentação teórica sobre os temas abordados, como sistema de transmissão de energia, demanda energética e a técnica de previsão de série temporal adotada, como também da metodologia adotada para obtenção dos resultados e análises.
 - Capítulo 3 - No capítulo três é apresentada a metodologia proposta para a execução do trabalho, como foi preparado os dados utilizados e configurações da rede neural utilizada.
 - Capítulo 4 - O quarto trás os resultados, o melhor modelo produzido a partir das técnicas utilizadas e suas representações gráficas.
 - Capítulo 5 - O quinto capítulo apresenta as conclusões finais do trabalho.
 - Capítulo 6 - Por fim em adicional, idéias para trabalhos futuros

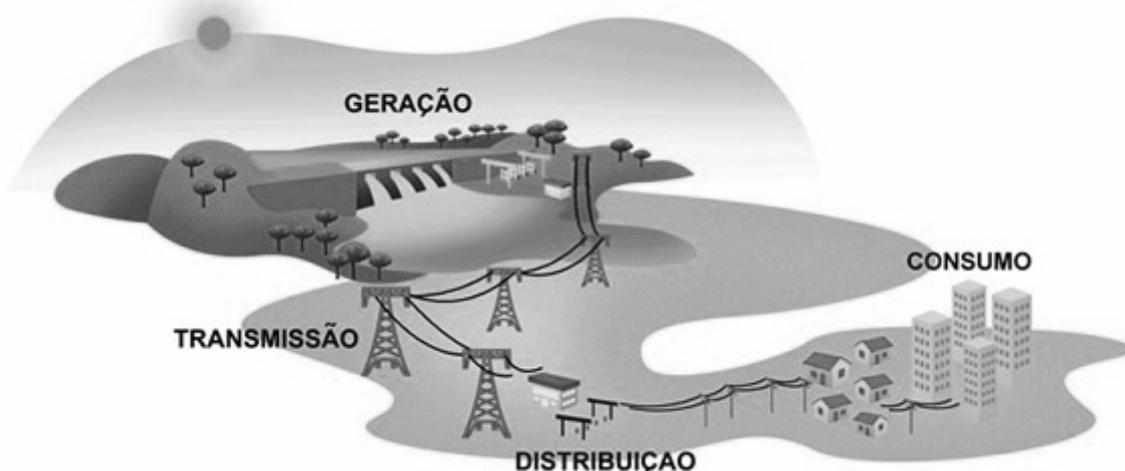
2 Fundamentação teórica

Neste capítulo será apresentado todo o embasamento teórico a respeito da previsão de demanda de energia elétrica, que é fundamental para conhecer a metodologia de sua previsão. A fundamentação teórica irá conter informações sobre o sistema elétrico de potência e seus subsistemas, sobre séries temporais e seus principais componentes, bem como sobre técnicas de previsão de séries temporais, com enfoque nas redes neurais artificiais, em especifica a LSTM que será a técnica de previsão de demanda utilizada neste trabalho. Por fim será abordado as métricas de avaliação dos resultados, seguido da técnica utilizada de *Dropout* a fim de evitar o *overfitting*, que é o aprendizado em excesso da rede neural em relação aos dados de treino.

2.1 Sistema Elétrico de Potência

De acordo com a Empresa de Pesquisa Energética (EPE, 2018), toda a energia disponível a atender uma determinada demanda de energia, seja de um país, estado ou local é chamada de matriz energética. Essa energia fornecida tem como origem um conjunto de fontes que fazem parte de uma matriz energética, que corresponde ao agrupamento das fontes à disposição de uma determinada região suprindo suas demandas. O conjunto de fontes disponíveis apenas para uma determinada geração de energia elétrica compõe a matriz elétrica, que juntas compõem a matriz energética.

Figura 1 – Representação de um sistema elétrico de potência.



Fonte: (ABRADEE, 2018)

Para suprir a demanda energética de um local é necessário um Sistema Elétrico de Potência (SEP), como cita Robba (ROBBA KAGAN, 2005) na qual sua principal função é fornecer energia elétrica aos usuários, com qualidade apropriada, de acordo com o que é demandado, convertendo energia de naturezas diversas, como hidráulica, mecânica, térmica, solar e entre outras em energia elétrica, distribuindo-as aos consumidores. Um sistema elétrico de potência pode ser subdividido em três grandes subsistemas como pode ser observado na figura 1, são eles: geração, transmissão e distribuição.

2.1.1 Subsistema: Geração

A geração é responsável pela conversão de diversas fontes de energia em energia elétrica. A sua geração se faz através de diversas fontes que compõem uma matriz energética. Como visto na Empresa de Pesquisa Energética (EPE, 2018) pode-se observar que a distribuição da matriz energética brasileira é composta predominantemente por fontes de energias renováveis, que chegam a representar 80% da produção interna de energia elétrica, levando em consideração a produção nacional e a importação de energia, que é essencialmente de origem renovável. De todas as fontes que compõem a matriz energética do Brasil, destaca-se a geração hidrelétrica, que corresponde a 56,8% da oferta interna de eletricidade, devido ao alto volume de recursos hídricos, distribuídos ao longo de todo o território brasileiro.

2.1.2 Subsistema: Transmissão

A energia produzida por um subsistema de geração necessita ser distribuída, e para isso é necessário subsistema de transmissão, que é responsável por elevar o nível de tensão da geração para níveis de tensão compatíveis com a transmissão. Este subsistema, é dividido em dois blocos. Um deles é o Sistemas Isolado, que recebe esse nome devido a condições geográficas das localidades em que são utilizados, que impedem a sua conexão ao outro sistema, que é denominado Sistema Interligado Nacional (SIN), que consiste em todo o sistema nacional de instalações elétricas do país.

2.1.3 Subsistema: Distribuição

O sistema de distribuição de energia elétrica no Brasil é regulado por um conjunto de resoluções da Agência Nacional de Energia Elétrica (ANEEL), que através das normas elaboradas nos Procedimentos de Distribuição (PRODIST) normatizam e padronizam as atividades técnicas relacionadas ao funcionamento e ao desempenho dos sistemas de distribuição de energia elétrica. A finalidade do PRODIST é disciplinar o planejamento de expansão e operação do sistema, a medição e a qualidade da energia elétrica, de forma

a garantir que os sistemas de distribuição operem com segurança, eficiência, qualidade e confiabilidade.

Conforme Robba (ROBBA KAGAN, 2005), o sistema de distribuição pode ser dividido em três pilares importantes:

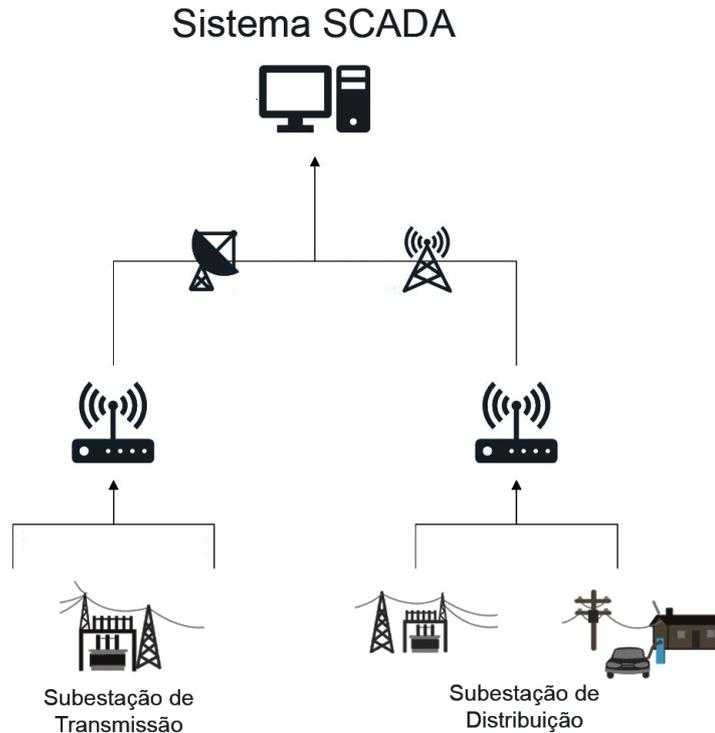
- Sistema de subtransmissão: Sua função é receber energia das estações de transmissão e transmiti-la às estações de distribuição e consumidores na tensão de subtransmissão, geralmente 138 kV ou 69 kV ou menos frequentemente 3,5 kV. Os consumidores de baixa tensão de transmissão são geralmente representados por grandes empreendimentos industriais, tratamento de água e estações de bombeamento.
- Sistema de distribuição primária: As redes de distribuição primária ou redes de média tensão provêm de subestações, que, no caso de redes aéreas, funcionam radialmente, para que quando as condições adversas se apresentem, seja possível transferir blocos de carga entre circuitos para atender às condições de operação, sendo em caso de manutenção corretiva ou preventiva. Essas redes normalmente operam na tensão de 13,8 kV atendendo consumidores primários, transformadores de distribuição e subestações, que alimentam a rede secundária ou de baixa tensão. Indústrias de médio porte, complexos comerciais, instalações de iluminação geral e etc. podem ser distinguidos dos principais consumidores, podendo ser aéreas ou subterrâneas, o primeiro é mais utilizado devido ao seu menor custo, e o segundo é bom para uso em áreas com maior densidade de carga, como no centro de uma grande cidade ou onde há restrições de paisagens.
- Sistema de distribuição secundária: Redes de distribuição secundárias, ou seja, média tensão, provêm de estações transformadoras, das quais a rede é de baixa tensão, 220/127 V ou 380/220 V, que pode funcionar em rede ou radialmente, fornecendo baixa tensão para consumidores, residências, pequenas empresas e indústrias. Atinge centenas de metros na qual destaca-se o predomínio dos consumidores domésticos.

Pode-se dizer que as subestações de distribuição, que rebaixam o nível de tensão para 13,8 kV e fornecem energia elétrica a rede de distribuição primária alimentam parte da rede de distribuição, dessa maneira, os circuitos de saída dessas subestações são denominados de alimentadores.

As subestações de distribuição de energia são monitoradas por um Sistema de Aquisição de Dados SCADA (*Supervisory Control and Data Acquisition*) que coleta dados de natureza elétrica (tensão, corrente, fator de potência, etc.), de diferentes pontos de medição localizados em equipamentos (disjuntores e religadores) e barramentos (MEDEIROS et al., 2016). Os dados obtidos nestas medições são enviados a um servidor que, por sua vez, armazena-os em um banco de dados. Essas medições, que são realizadas ao longo do

tempo com periodicidade de 15 minutos, são denominadas séries temporais. Na Figura 2, é ilustrada a conexão entre as medições nas subestações à um sistema SCADA.

Figura 2 – Sistema de aquisição de dados.



Fonte: Elaborado pelo autor 2022

2.2 LSTM

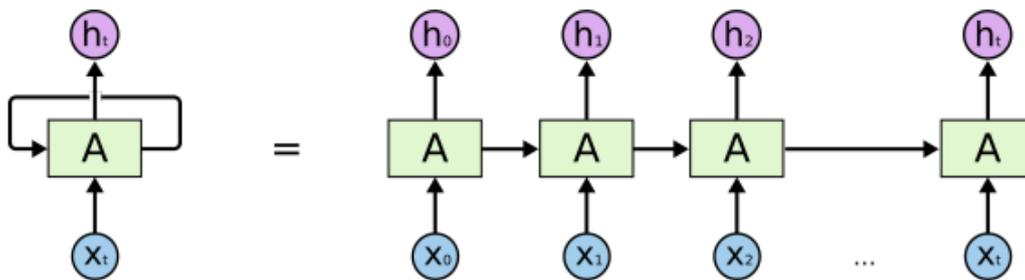
Redes Neurais Recorrentes (RNNs) conseguem utilizar suas conexões de feedback junto com entradas atuais para poder melhor prever valores futuros, que podemos definir como a memória de curto prazo, já a memória de longo prazo utiliza de pesos e bias que mudam lentamente ao longo do tempo. Segundo Hochreiter (HOCHREITER; SCHMIDHUBER, 1997), tais algoritmos são muito utilizados para controles não Markovianos e composição musical, mas levam muito tempo para processar e não tem um bom desempenho, em especial quando as entradas de treinamento são grandes, e acabam resultando em erros muito grandes indo para $\pm\infty$.

A partir disso surge a rede LSTM (*Long Short Term Memory*), uma rede neural recorrente que utiliza um método de aprendizado em gradiente, eliminando o problema de erros quando a entrada da rede contem ruídos ou são irrelevantes para o aprendizado, sem perdas ou intervalo mínimo de dados, impedindo ainda que o resultado não exploda ou suma para o infinito devido a suas funções internas sigmóides e tangentes hiperbólicas.

2.2.1 Estrutura interna da LSTM

Em *deeplearning*, as RNNs são Redes Neurais Artificiais, que, diferentemente das redes neurais *feedforward*, usam conexões reversas, onde os nós podem se conectar a outros nós nas camadas anteriores, ou a si mesmos, formando um ciclo retroalimentado. Como resultado, essas arquiteturas tem capacidade de memória, onde o valor passado do neurônio, juntamente com a entrada da camada anterior, são a entrada do neurônio/camada. Portanto, os valores de saída passados do neurônio, são determinados pelas entradas passadas, influenciando sua saída atual.

Figura 3 – Arquitetura básica da rede neural recorrente.



Fonte: (COLAH, 2015)

Na Figura 3 se ilustra a estrutura de uma rede neural recorrente, que, a esquerda se retroalimenta, mas que se assemelha a uma rede em corrente replicada, que passa a informação processada por cada uma célula adiante, na qual os processos internos serão explicados mais a frente.

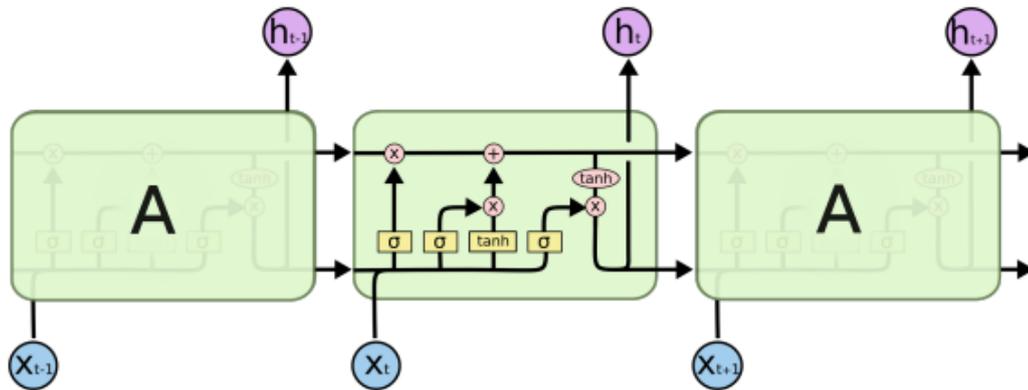
As RNNs são frequentemente utilizadas para reconhecimento de padrões quando os resultados anteriores influenciam os resultados atuais, como dados de series temporais ou tradução. Por outro lado, as redes neurais recorrentes simples, com a arquitetura exibida acima, sofrem com o problema de gradiente de fuga, o que significa que a rede só pode se lembrar de entradas recentes e rapidamente esquece mais entradas de longo prazo. Para lidar com esse problema, foi introduzida uma variante do RNN conhecida como redes de memória de longo prazo, ou LSTMs (HOCHREITER; SCHMIDHUBER, 1997).

Esse tipo de rede alcança a capacidade de memória de longo prazo com uma nova arquitetura que é ilustrada na Figura 4. Nesta arquitetura, cada módulo repetitivo (neurônio) mantém um estado e possui 3 portas de entrada ou comumente chamado de portões, no qual cada um afeta o estado da célula de uma maneira diferente:

- Portão de entrada: decide quanto atualizar cada valor no estado da célula com os novos valores de entrada, ou seja, controla se a célula vai adquirir um novo conhecimento vindo da célula anterior;

- Portão de saída: filtra a saída com base no estado da célula e decide se a próxima célula irá obter os dados processados na célula atual;
- Portão de esquecimento: decide quanto esquecer (apagar) cada valor no estado da célula atual, esse esquecimento pode variar sendo multiplicado por uma constante entre 0 e 1.

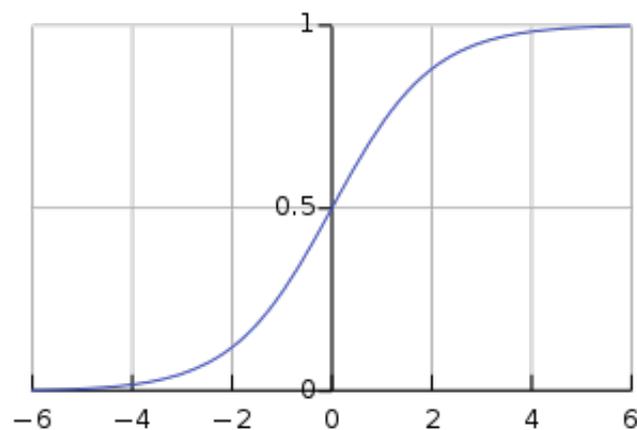
Figura 4 – Arquitetura de rede de memória de longo prazo.



Fonte: (COLAH, 2015)

Esses portões são geralmente compostos por uma função de ativação sigmoide, que gera um valor entre 0 e 1 como se ilustra Figura 5, descrevendo o nível de ativação do portão (onde 0 está completamente fechado e 1 está completamente aberto). Devido a sua eficácia os LSTMs tornaram-se muito populares, bem como a forma dominante de RNN usada em propósitos práticos, particularmente no processamento de dados de sequência e series temporais.

Figura 5 – Plot da função sigmoide.

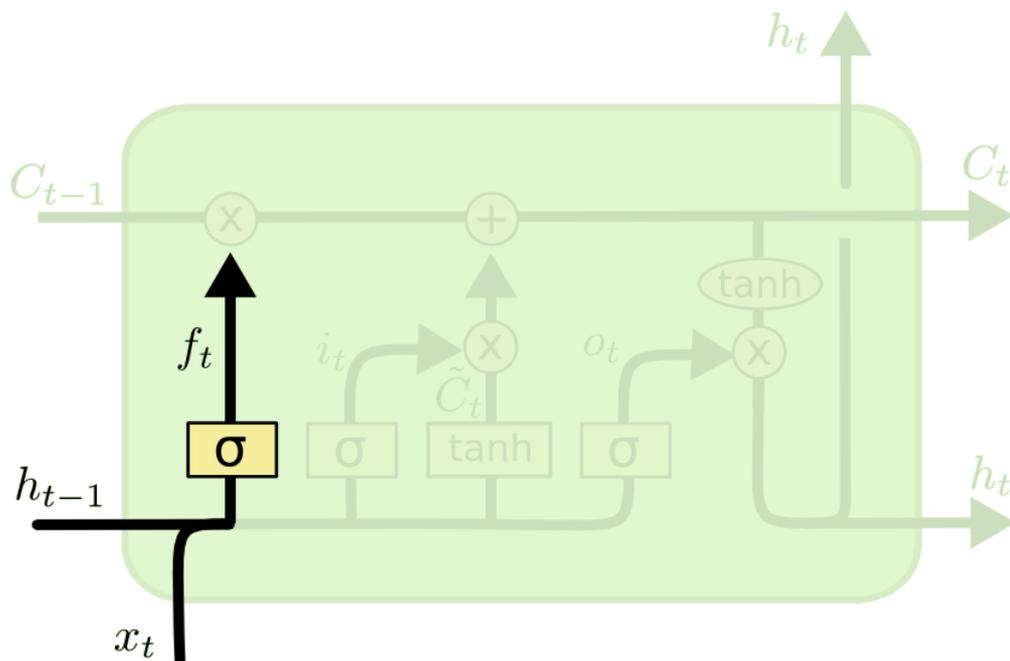


Fonte: Qef, 2008

2.2.2 Passo a passo da LSTM

A primeira parte de uma LSTM é responsável pelo descarte de informações da célula anterior e de dados que são inseridos na célula atual. O esquecimento da informação é ajustado por uma função sigmoide "σ", que varia seu valor entre 0 e 1. Um sigma com valor 0 representa que a entrada será totalmente esquecida, enquanto um sigma com valor 1 representa que a entrada será totalmente lembrada. A entrada h_{t-1} simboliza os dados provindos da célula anterior, do tempo t-1, que é concatenada com a entrada x_t , que é fornecida pelo usuário.

Figura 6 – Primeira parte da LSTM



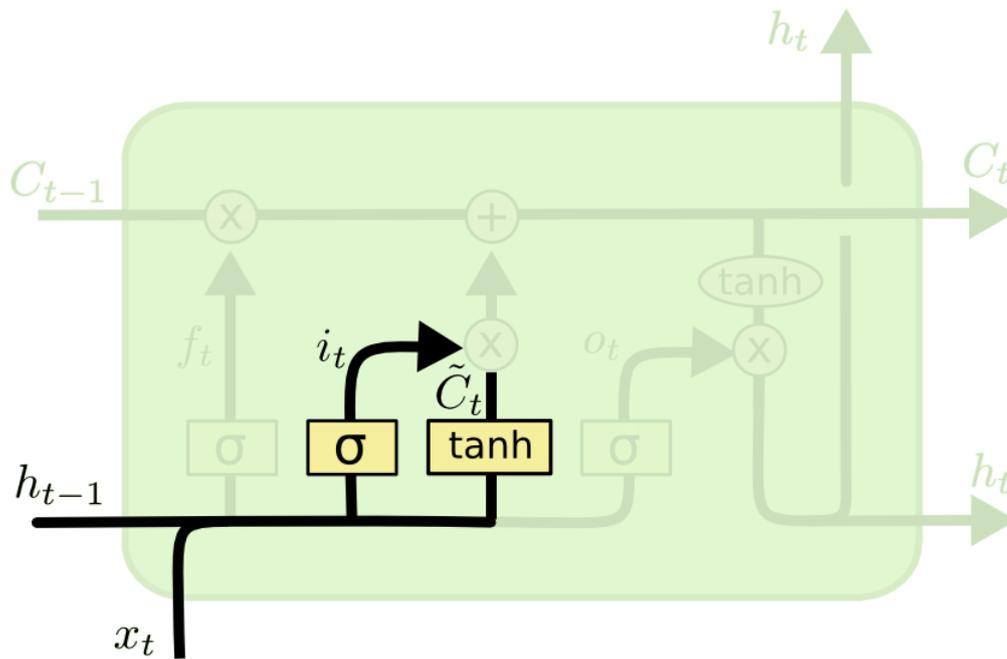
Fonte: (COLAH, 2015)

Pela função que representa a primeira parte podemos ver a importância da função sigmoide "σ", que multiplica W_f , que são os pesos, seguido da entrada h_{t-1} provinda da célula anterior, somado ao bias.

$$f_t = \sigma(W_f \cdot [h_{(t-1)}, x_t] + b_f) \quad (2.1)$$

A segunda parte é decidir quais novas informações vamos armazenar no estado da célula. Isso tem duas partes. Primeiro, uma camada sigmoide chamada "camada de entrada" decide quais valores atualizar. A camada tanh (tangente hiperbólica) então cria um vetor de novos valores candidatos para mover para a célula atual C_t . Na próxima etapa, combinaremos os dois para criar uma atualização para a célula atual.

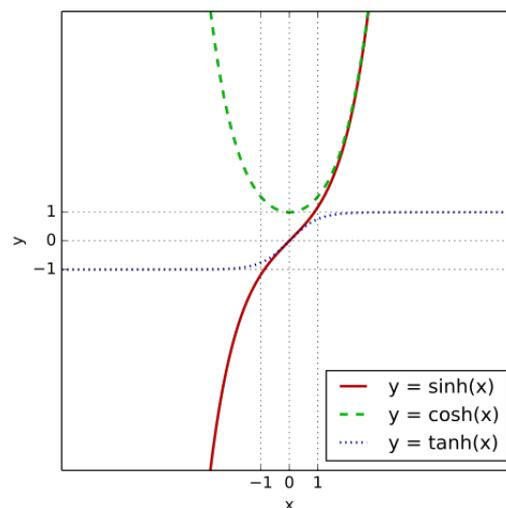
Figura 7 – Segunda parte da LSTM



Fonte: (COLAH, 2015)

Nessa etapa vemos que a função i_t se comporta como a etapa anterior, f_t , multiplicado pela função sigmoide, com peso multiplicando a entrada provinda da célula anterior e a entrada atual da célula, somado ao bias. Já a função \tilde{C}_t conta com a tangente hiperbólica no lugar da função sigmoide. Ela é responsável por não deixar os valores irem para $\pm\infty$, pois assume o valor entre -1 e 1.

Figura 8 – Plot da função Tangente Hiperbólica



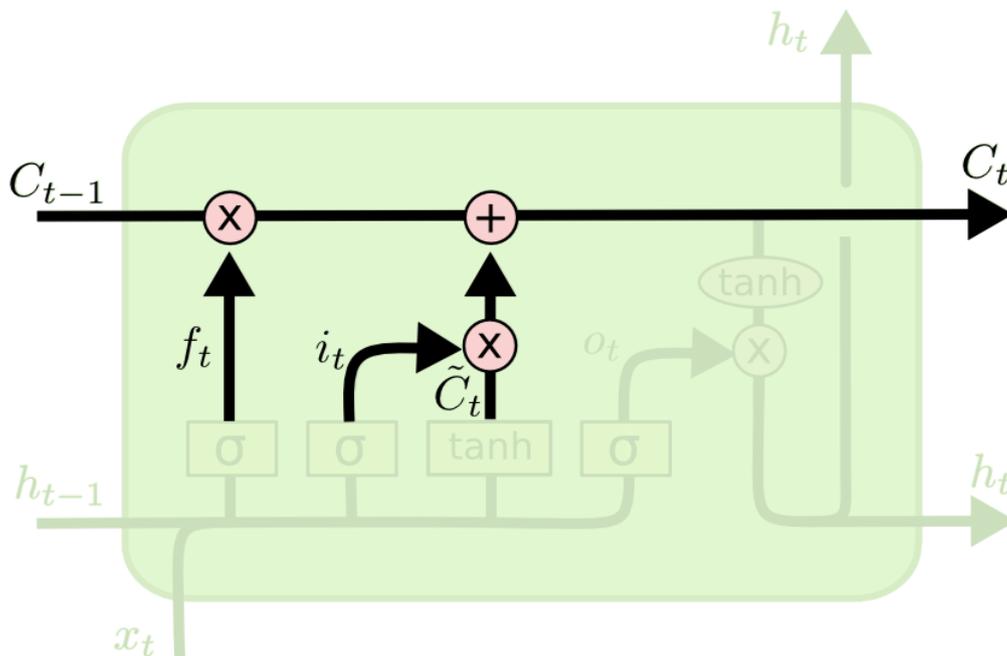
Fonte: Fylwind, 2016

$$\begin{aligned} i_t &= \sigma(W_i \cdot [h_{(t-1)}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_C \cdot [h_{(t-1)}, x_t] + b_C) \end{aligned} \quad (2.2)$$

Agora é hora de atualizar o estado da célula antiga, C_{t-1} , no novo estado da célula C_t . Os passos anteriores já decidiram o que fazer, só precisamos realizar a operação para alterar os valores da célula.

Multiplicamos o estado antigo com f_t , esquecendo as coisas que escolhemos no estado anterior. Então adicionamos $i_t * C_t$ e esses serão os novos valores dimensionados pelo o quanto decidimos atualizar do valor de cada estado.

Figura 9 – Terceira parte da LSTM

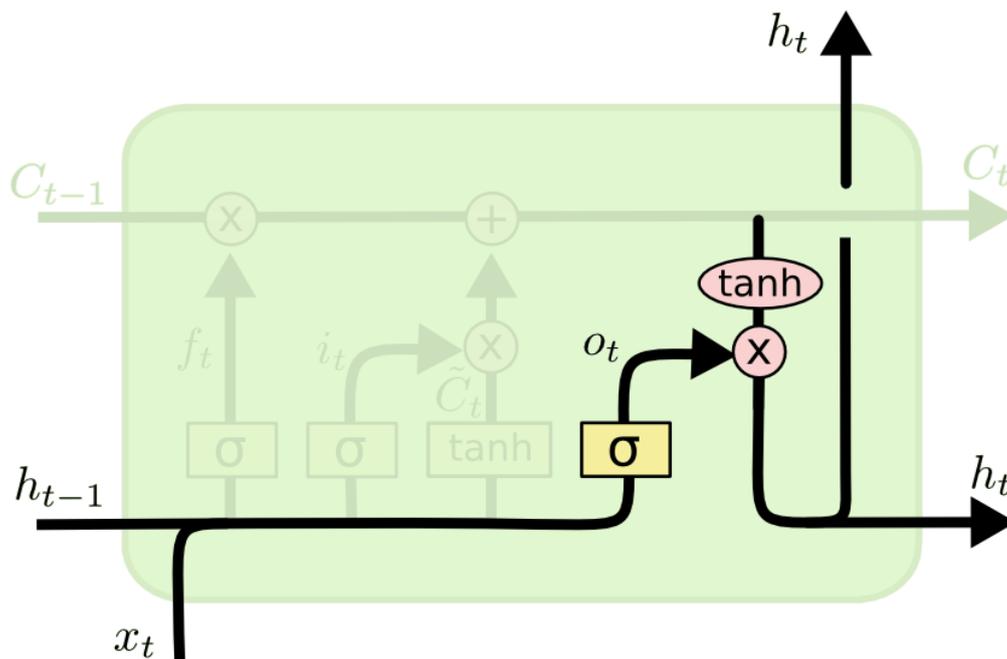


Fonte: (COLAH, 2015)

$$C_t = f_t * C_{(t-1)} + i_t * \tilde{C}_t \quad (2.3)$$

Por último, decidiremos o que vai prosseguir para a próxima célula. Essa saída será baseada em nosso estado de célula atual, mas será uma versão filtrada das anteriores. Primeiro, percorremos por uma camada sigmoide que decide quais partes do estado da célula anterior e da entrada atual vamos produzir. Então, nós colocamos o estado da célula através de tangente hiperbólica (para empurrar os valores para estar entre -1 e 1) e multiplicamos pela saída do sigmoide, de modo que apenas produzamos as partes que decidimos.

Figura 10 – Quarta parte da LSTM



Fonte: (COLAH, 2015)

$$\begin{aligned} o_t &= \sigma(W_o \cdot [h_{(t-1)}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \quad (2.4)$$

2.3 Métricas de avaliação

O processamento de uma rede neural pode ser dividido em duas fases: aprendizado (*learning*) e execução (*recall*). O treinamento ou aprendizado é o processo de atualização dos pesos sinápticos. Depois da rede estar devidamente treinada, quando o erro estiver no nível satisfatório, seus pesos sinápticos são armazenados em uma matriz peso (W). Na fase de execução, a rede é utilizada apenas no modo progressivo (*feed-forward*), ou seja, novos registros são apresentados, eles são processados nas camadas intermediárias e os resultados são apresentados na camada de saída.

Após o treinamento, diferentes métodos de ensino devem ser avaliados para que possam ser comparados. Algumas métricas comumente utilizadas em problemas de classificação são utilizadas para realizar a avaliação. Três dessas métricas são utilizadas neste trabalho e desenvolvidas a seguir.

2.3.1 Erro médio quadrático (MSE)

Na área de probabilidade e estática, segundo (PISHRO-NIK, 2016) temos que o erro médio quadrático (MSE) ou desvio médio quadrático (MSD) de um valor estimado mede a média dos seus erros, que é, por definição, a distância euclidiana entre o ponto previsto e o valor real, como vemos na equação abaixo:

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2. \quad (2.5)$$

Como essa distância é calculada com base na média e no quadrado do seu valor estimado, torna-se uma função de risco, pois como seu valor sempre será positivo e não nulo, acaba-se perdendo valores potencialmente importantes.

Em *machine learning*, especificamente na minimização do risco empírico, o MSE é uma medida de qualidade de um estimador. Quanto menor a distância euclidiana, mais próximo de zero será esse valor e menor será o seu erro.

2.3.2 Mean absolute percentage error (MAPE)

Outro exemplo de métricas de avaliação de redes neurais é o erro percentual médio absoluto (MAPE). Sendo x um vetor de variáveis e é usado como entrada para um modelo de regressão, y seria a variável alvo e g um modelo de regressão, o MAPE de g seria obtido pela média da razão $|g(x)y|$ por $|y|$. O MAPE é frequentemente utilizado na prática devido à sua interpretação muito intuitiva em termos de erro relativo. A utilização do MAPE é relevante em finanças, por exemplo, como ganhos e perdas são frequentemente medidos em valores relativos. Também é útil para calibrar os preços dos produtos, já que os clientes às vezes são mais sensíveis a variações relativas do que variações absolutas. Em aplicações do mundo real, o MAPE é frequentemente utilizado quando a quantidade a se prever normalmente se mantém acima de zero. De forma mais geral, segundo Armstrong (ARMSTRONG; COLLOPY, 1992), MAPE é uma métrica adequada para aplicações de previsão, especialmente em situações onde os dados que serviram para treinar a rede neural são suficientemente grandes.

$$\text{MAPE} = \frac{100\%}{n} \sum_{t=1}^n \left| \frac{A_t - F_t}{A_t} \right| \quad (2.6)$$

Acima vemos a sua equação, onde A_t é o valor real e F_t é o valor previsto. Sua diferença é dividida pelo valor real A_t . O valor absoluto desta razão é somado para cada ponto previsto no tempo e dividido pelo número de pontos n .

2.3.3 Erro máximo

O erro máximo é um indicador da precisão de uma estimativa e é definido como a diferença de medida entre o valor real e o valor estimado.

O erro neste caso é a diferença entre o valor medido e o valor real, que é uma forma de considerar o erro ao medir a precisão numérica.

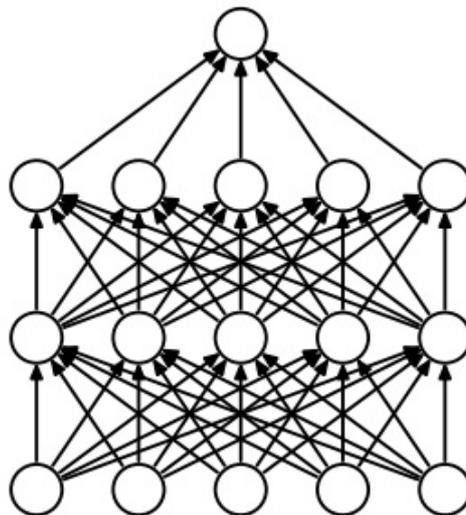
Se você conhece o valor medido e o valor atual, pode calcular o erro absoluto simplesmente subtraindo-os. No entanto, às vezes você pode não saber o valor real.

Neste caso, você precisa usar o erro máximo possível como o erro absoluto. Se você souber o valor verdadeiro e o erro relativo, poderá calcular o erro absoluto.

2.4 Dropout

As redes neurais profundas contêm múltiplas camadas ocultas não lineares e isso as torna um modelo complexo, podendo aprender relacionamentos mais complicados entre suas entradas e saídas. Com os dados de treinamento limitados, no entanto, muitos desses relacionamentos complexos serão o resultado do ruído de amostragem, pois eles existem no conjunto de treinamento, mas não nos dados reais de teste, mesmo que sejam extraídos da mesma base.

Figura 11 – Rede neural padrão sem o *Dropout*



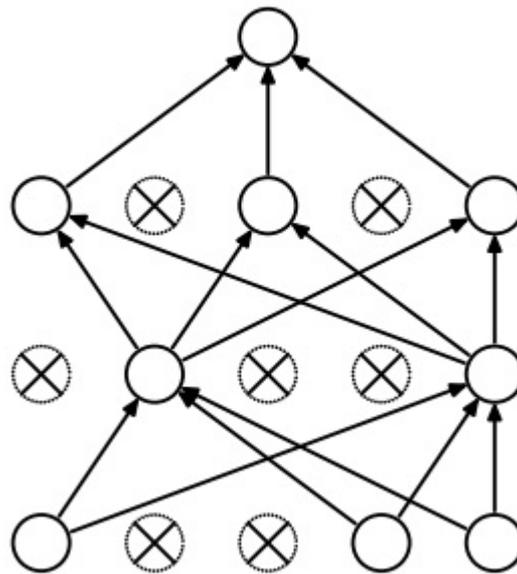
Fonte: (SRIVASTAVA et al., 2014)

Isso acaba levando ao *overfitting* e muitos métodos foram desenvolvidos para reduzi-lo. Isso inclui interromper o treinamento assim que desempenho em um conjunto de validação começa a piorar, ou não melhorar, introduzindo penalidades na variável de peso

de vários tipos como regularização L_1 e L_2 e compartilhamento de peso suave (HINTON; NOWLAN, 1994).

Como o nosso poder de computação é limitada, a melhor maneira de "regularizar" um modelo de tamanho fixo é que se faça uma média das previsões de todas as configurações possíveis dos parâmetros, ponderando cada configuração por suas probabilidades posteriores dado a base de treinamento. Isso às vezes pode se aproximar bastante do resultado real, sendo muito bom para modelos simples ou pequenos (XIONG; BARASH; FREY, 2011).

Figura 12 – Rede neural padrão



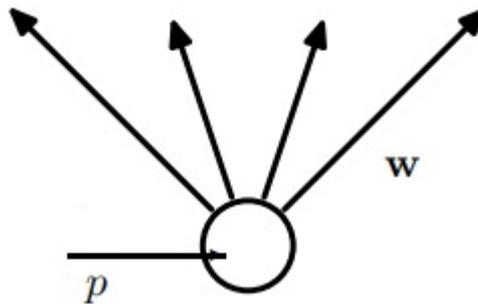
Fonte: (SRIVASTAVA et al., 2014)

A combinação de modelos quase sempre melhora o desempenho dos métodos de *machine learning*. Com grandes redes neurais, no entanto, a ideia óbvia de calcular a média das saídas de muitas redes treinadas separadamente é claramente um processo que consumiria tempo. Combinar vários modelos é mais efetivo, sabendo que os modelos individuais são diferentes uns dos outros e para tornar modelos de redes neurais diferentes, eles devem ter arquiteturas diferentes ou ser treinados em dados diferentes. Treinar muitas arquiteturas diferentes é difícil porque encontrar o melhor hiper parâmetros para cada arquitetura é uma tarefa trabalhosa e treinar cada grande rede requer muitos cálculos. Sabendo que, quando estamos trabalhando com grandes redes, elas requerem uma quantidade de dados maior para treinamento e pode não haver dados disponíveis suficientes para treinar diferentes redes em diferentes subconjuntos de dados. Mesmo que alguém fosse capaz de treinar muitas grandes redes diferentes, usá-las só para teste é inviável em aplicações onde é importante ter uma resposta de baixa latência.

Dropout é uma técnica que aborda essas duas questões. Evita o excesso de ajustes

e fornece uma maneira de combinar muitas redes neurais diferentes estruturadas de forma eficiente. O termo do inglês "abandono" refere-se ao abandono de unidades (ocultas e visíveis) em uma rede neural. Ao descartar uma unidade, queremos dizer iremos removê-la temporariamente da rede, juntamente com todas as suas conexões de entrada e saída, conforme mostrado na Figura 12. A escolha de quais unidades esquecer é aleatória. No caso mais simples, cada unidade é mantida com uma probabilidade constante p independente das outras unidades, onde p pode ser escolhido usando uma validação definida ou pode simplesmente ser definido como 0.5, o que parece ser próximo do ideal para uma ampla gama de redes e aplicações.

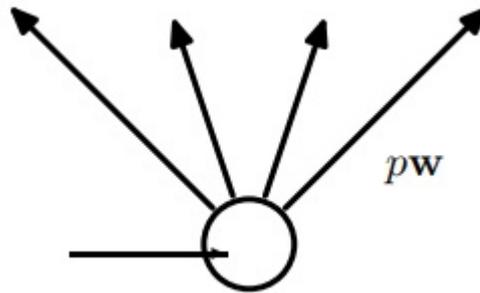
Figura 13 – Camada em treino com a probabilidade p e saída com pesos W .



Fonte: (SRIVASTAVA et al., 2014)

Aplicar *Dropout* a uma rede neural equivale a amostrar uma rede "reduzida" dela. A rede reduzida consiste em todas as unidades que sobreviveram ao esquecimento 13. Uma rede neural com n unidades, pode ser vista como uma coleção de 2^n possíveis redes neurais reduzidas. Todas essas redes compartilham pesos para que o número total de parâmetros ainda seja $O(n^2)$, ou menos. Para cada apresentação de cada caso de treinamento, uma nova rede reduzida é amostrada e treinada. Portanto, treinar uma rede neural com *Dropout* pode ser visto como treinar uma coleção de 2^n redes reduzidas com amplo compartilhamento de peso entre si, onde cada rede reduzida é treinada muito raramente, ao todo.

Figura 14 – Na etapa de teste, a camada com os pesos sendo multiplicados por p .



Fonte: (SRIVASTAVA et al., 2014)

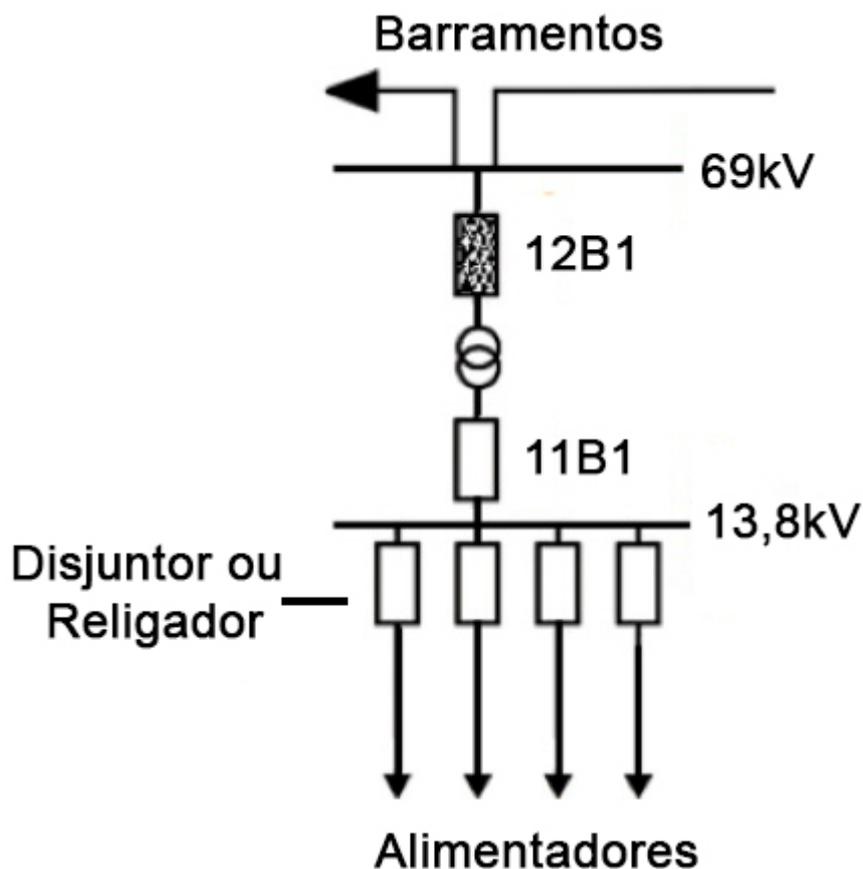
No entanto, um método muito simples e que funciona bem na prática é o da média aproximada. A ideia é usar uma única rede neural na hora do teste sem *Dropout*. Os pesos desta rede serão versões reduzidas dos pesos treinados. Se uma unidade for retida com probabilidade p durante o treinamento, os pesos de saída dessa unidade são multiplicados por p durante o período de teste, conforme é mostrado na Figura 13 e Figura 14. Isso garante que, para qualquer unidade oculta, a saída esperada (sob o método utilizado para descartar unidades durante o período de treinamento) é a mesma que a saída real durante o período de teste. Ao fazer esse escalonamento, 2_n redes com pesos compartilhados podem ser combinadas em uma única rede neural e ser utilizada no período de teste. Segundo Srivastava (SRIVASTAVA et al., 2014), treinar uma rede com *Dropout* e usar este método de média aproximada no momento do teste dos dados, leva a um erro significativamente menor de generalização em uma ampla variedade de problemas de classificações, quando comparado com o treinamento com outros métodos de regularização.

3 Metodologia Proposta

3.1 Apresentação do estudo de caso

O objeto de estudo escolhido para este trabalho foi de uma subestação de uma concessionária de energia elétrica, na qual os dados foram fornecidos pela mesma. As medidas obtidas da subestação foram aferidas em um período de 5 anos, desde 2008 até 2013. Os pontos de medição, cujos dados de natureza elétrica foram obtidos, estavam referidos aos barramentos (69kV e 13,8kV) e equipamentos (disjuntores ou religadores) de um dos seus alimentadores, 12B1, na qual podemos ver representado no diagrama unifilar abaixo.

Figura 15 – Diagrama unifilar

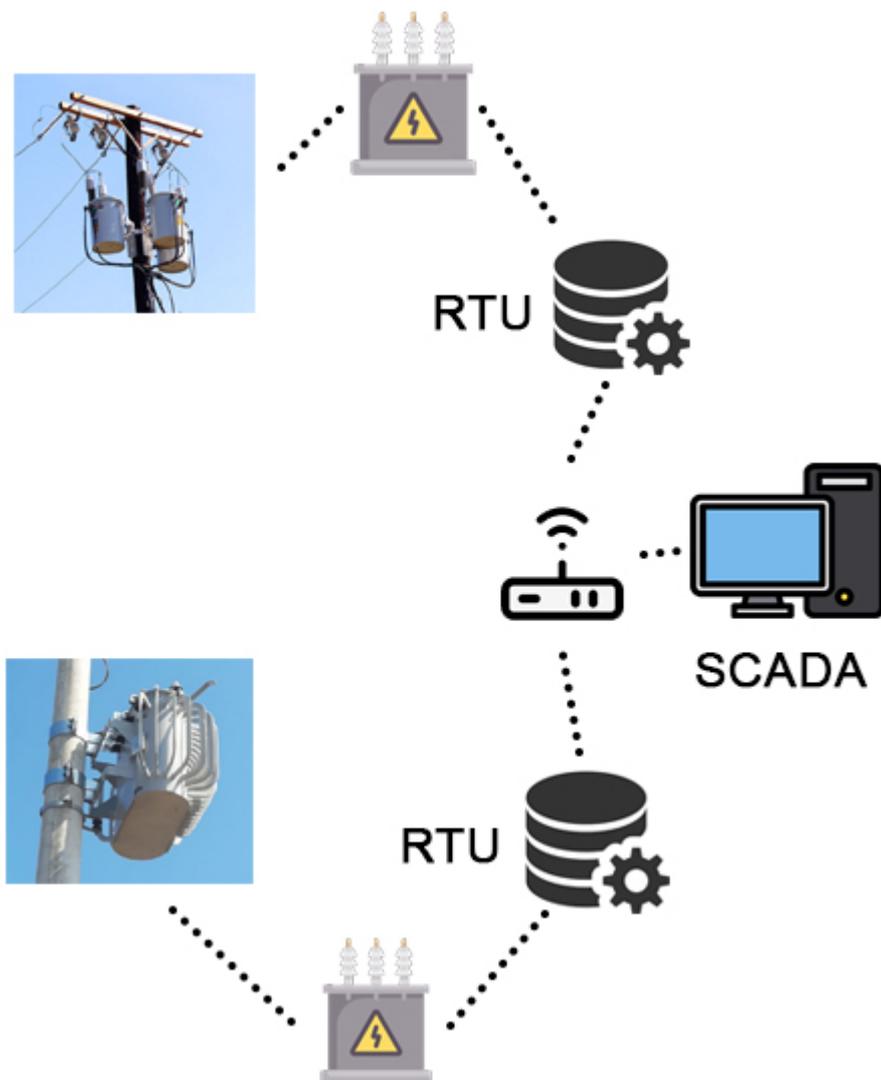


Fonte: Elaborado pelo autor 2022

Em uma subestação, todas as medidas elétricas, tanto como potência, corrente e tensão são feitas por equipamentos medidores específicos para cada grandeza. Todas as

medidas passam por um sistema de aquisição de dados chamado RTU (Unidade Terminal Remota), que as registram e enviam para um servidor de dados, com base em um sistema SCADA, como ilustrado na figura abaixo.

Figura 16 – Esquema dos elementos básicos de um monitoramento de subestações e sistema SCADA

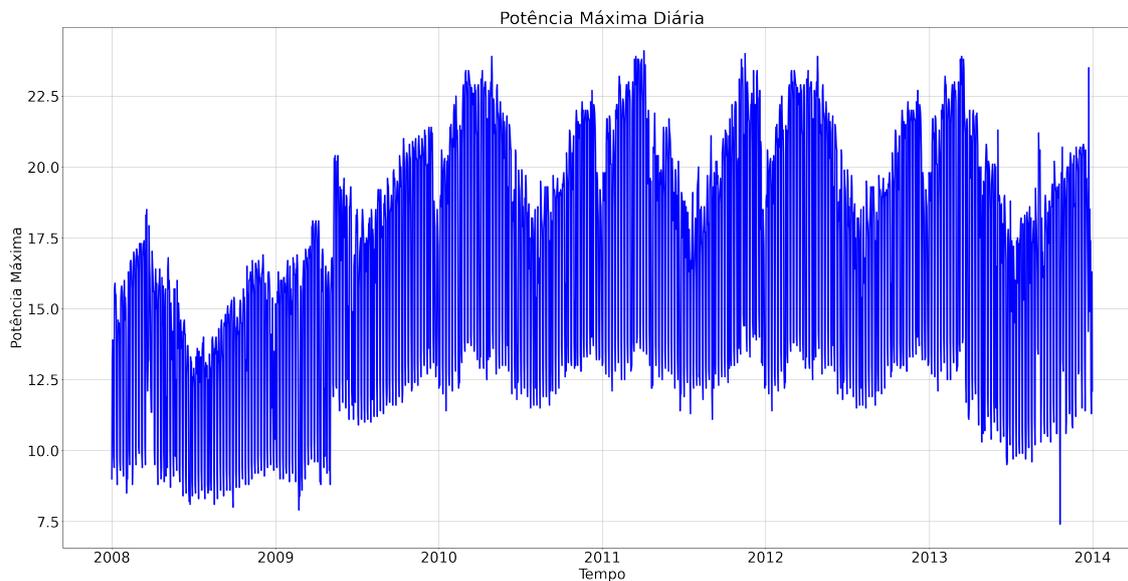


Elaborado pelo autor 2022

A medições após serem extraídas são utilizadas para análise de comportamento das redes, a fim de, tanto melhorar seu desempenho quanto melhorar seu nível de segurança. Essas medições, que serão chamadas de series temporais, são extraídas em um intervalo de tempo de 15 minutos. Observando os dados aferidos abaixo é possível analisar que nos perfis de demanda há uma alta não-linearidade da série histórica, com características

sazonais e tendência.

Figura 17 – Série temporal de demanda para o período de 2008-2013



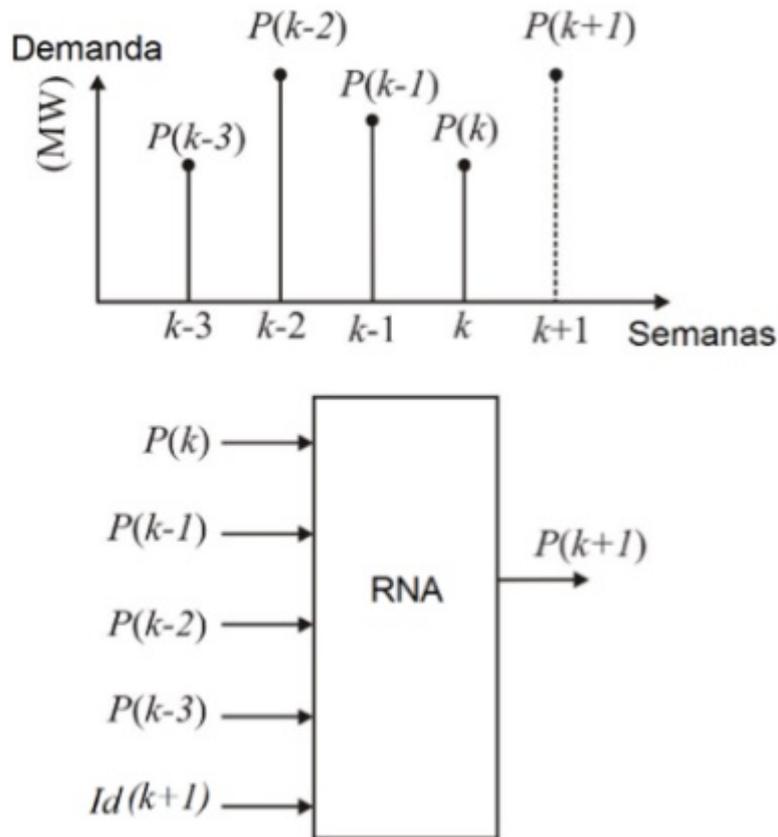
Fonte: Elaborado pelo autor 2022

3.1.1 Modelagem do Banco de dados

Para a modelagem da previsão de demanda energética, foram utilizados softwares como o Visual Studio Code®, Google Colab® e Microsoft Excel®. Todos os dados de potência presentes neste trabalho foram fornecidos pela concessionária de energia elétrica, e para sua utilização foi feito um tratamento prévio para remoção de outliers e definição de métricas semanais para facilitar as análises e desempenho da rede neural.

Para as entradas da rede neural artificial, considerou-se cinco valores de potência, sendo valor atual e valores passados. Na Figura 18 é apresentado um modelo representativo das entradas de variáveis da rede neural artificial e suas saídas, onde k significa o instante atual, $k-n$ são valores passados e $k+n$ são os valores futuros, estes últimos sendo os resultados da nossa previsão de demanda.

Figura 18 – Modelo de entradas e saídas da RNA



Fonte: (MEDEIROS et al., 2016)

3.1.2 Previsão de curto prazo de máxima semanal usando LSTM

Para as simulações, foram utilizados, tanto o software Visual Studio Code® quanto do Google Colab®, construindo uma RNA utilizando a técnica de LSTM, os seguintes parâmetros e configurações iniciais foram considerados:

Número de camadas: 4 camadas;

A primeira camada com 64 nós de LSTM

A segunda camada Dense com 32 nós e ativação Relu

A terceira camada Dense com 32 nós e ativação Relu

A quarta camada Dense com 1 nó para a saída.

Usando as linhas de código abaixo, foi configurado o modelo da LSTM:

Figura 19 – Modelo Sequencial

```
model = Sequential([layers.Input((4, 1)),  
                    layers.LSTM(64),  
                    layers.Dense(32, activation='relu'),  
                    layers.Dense(32, activation='relu'),  
                    layers.Dense(1)])
```

Fonte: Elaborado pelo autor 2022

O modelo sequencial tem a primeira camada de entrada, sendo descrita como "layers.Input" na qual foram definidas a partir do banco de dados, sendo 4 entradas e 1 saída. Os 4 dados de entradas servirão para treinar a rede neural e o dado de saída servirá para comparação com a previsão da rede.

A segunda camada de LSTM tem 64 neurônios. A entrada que eram 4 da camada passa agora se multiplica para 64 entradas diferentes.

A terceira camada e a quarta são idênticas. Recebendo a entrada da camada 2, reduz os dados para a dimensão de 32, as duas com a ativação ReLu.

A função de ativação linear retificada ou ReLU para abreviar é uma função linear por partes que produzirá a entrada diretamente se for positiva, caso contrário, produzirá zero. Tornou-se a função de ativação padrão para muitos tipos de redes neurais porque um modelo que a utiliza é mais fácil de treinar e geralmente obtém melhor desempenho (BROWNLEE, 2019).

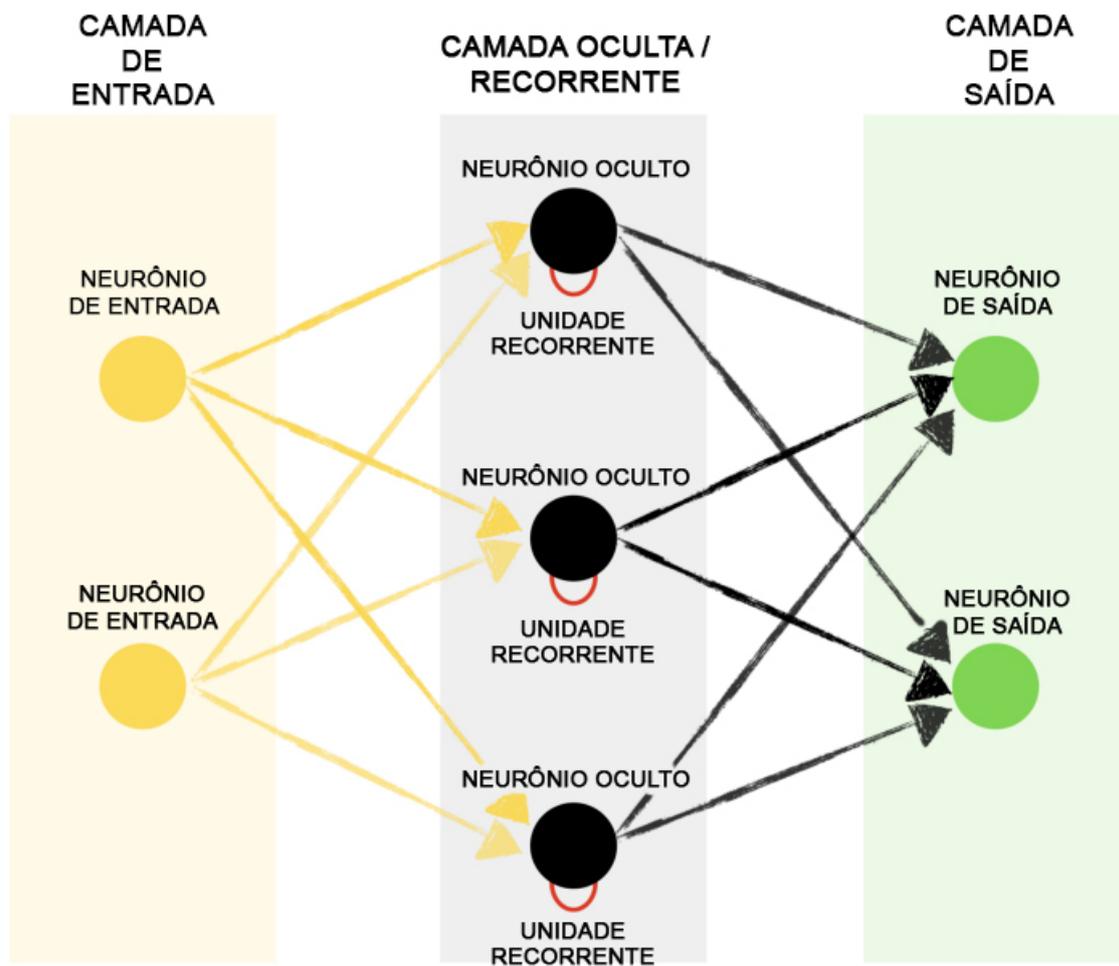
Finalmente a última camada com 1 nó, diminuindo a dimensão dos dados da camada anterior para 1 saída.

As camadas que contém o argumento "Dense" são camadas densas, que, em qualquer rede neural, é uma camada que está profundamente conectada com sua camada anterior, o que significa que os neurônios da camada estão conectados a todos os neurônios de sua camada anterior. Essa camada é a camada mais comumente usada em redes de redes neurais artificiais (SVERMA, 2022).

O modelo otimizador utilizado foi o Adam com a taxa de aprendizado de 0.001.

A otimização de Adam é um método de descida de gradiente estocástico baseado na estimativa adaptativa de momentos de primeira e segunda ordem. De acordo com Kingma (KINGMA; BA, 2014), o método é "computacionalmente eficiente, requer pouca memória, é invariante ao reescalonamento diagonal de gradientes e é adequado para problemas grandes em termos de dados/parâmetros".

Figura 20 – Modelo LSTM



Fonte: Elaborado pelo autor 2022

Para ser definido o número de neurônios utilizados na rede neural foi preciso uma iteração para abranger todos os valores possíveis de configurações, mas pela limitação computacional não foi possível realizar todas.

A princípio seriam feitas combinações com 64 nós da primeira camada, seguido de 32 da segunda camada junto com 32 da terceira camada, fazendo uma análise combinatória, chega-se a um total de 65.536 combinações de neurônios diferentes dentro da rede. Porém, utilizando o desempenho de um *Desktop* com as configurações de um processador Intel® i7-2600 3.40GHZ e 8gb de memória *RAM* foi alcançado o estimado de aproximadamente 15 segundos para treinar, avaliar e testar cada configuração. Somando todo o tempo que levaria para calcular as 65mil combinações, daria em torno de um total de 8 dias de processamento de dados, que no caso seria impraticável para o momento.

Utilizando o poder computacional de computadores do Google Colab® foi encontrado em média o mesmo tempo de 10 segundos para processamento de cada configuração desejada.

Por fim foi decidido utilizar do método de iteração, mas diminuindo o número de possibilidades, usando um *step* de 5 em 5, diminuindo a quantidade de combinações para 432.

3.1.3 Metodologia

Para iniciar o ambiente de trabalho com a LSTM é necessário importar algumas bibliotecas. As bibliotecas utilizadas neste trabalho foram as mencionadas abaixo.

Figura 21 – Bibliotecas

```
import pandas as pd
from datetime import datetime
import matplotlib.pyplot as plt
from sklearn.preprocessing import MinMaxScaler
import numpy as np
from keras.layers import Dropout
from keras.models import Sequential
from keras.optimizers import Adam
from keras import layers
from sklearn.metrics import mean_squared_error
from sklearn.metrics import mean_absolute_percentage_error
from sklearn.metrics import max_error
```

Fonte: Elaborado pelo autor 2022

Para o uso do banco de dados fornecido pela concessionária de energia elétrica, foram necessários alguns ajustes dentro do código utilizado.

Primeiro foi preciso separar os dados obtidos em dados semanais, para que pudesse ser feito todo o tratamento na base, incluindo separar os dados em dias e semanas, para melhorar o desempenho da rede neural. Após ser filtrado em dias, foi separado os máximos diários, pois o planejamento para alta demanda precisa ser feito com base no máximo requisitado. Em seguida, com os dados de máximas diárias, foram coletados os máximos semanais. Com os máximos semanais filtrados, a tabela de dados foi moldada para um modelo de janela, onde a quatro primeiras colunas da tabela seriam a entrada da rede neural, contendo os dados filtrados anteriormente, e a última coluna seria para a parte de teste da rede neural. Após o processamento da tabela, sua estrutura chegou a esse modelo abaixo:

	TEMPO	POTENCIA	POTENCIA(K-1)	POTENCIA(K-2)	POTENCIA(K-3)	POTENCIA(K+1)
	max	max	max	max	max	
0	27/01/2008	15.800.000	14.600.000	15.900.000	13.900.000	16.000.000
1	03/02/2008	16.000.000	15.800.000	14.600.000	15.900.000	16.299.999
2	10/02/2008	16.299.999	16.000.000	15.800.000	14.600.000	16.700.001
3	17/02/2008	16.700.001	16.299.999	16.000.000	15.800.000	17.000.000
4	24/02/2008	17.000.000	16.700.001	16.299.999	16.000.000	17.100.000
...
305	01/12/2013	20.700.001	20.400.000	20.500.000	20.000.000	20.700.001
306	08/12/2013	20.700.001	20.700.001	20.400.000	20.500.000	20.799.999
307	15/12/2013	20.799.999	20.700.001	20.700.001	20.400.000	20.600.000
308	22/12/2013	20.600.000	20.799.999	20.700.001	20.700.001	23.500.000
309	29/12/2013	23.500.000	20.600.000	20.799.999	20.700.001	16.299.999

Após o processamento da tabela foi decidido dividir o estudo em 3 partes, a primeira parte seria avaliada a necessidade de utilização de um recurso da rede LSTM que se chama "Validação", onde os dados que foram utilizados para treinar a rede neural também é utilizado para validar o treinamento, com intuito de diminuir o erro de sua previsão. A segunda parte é a verificação da eficiência da normalização dos dados de entrada da rede neural, onde os dados são escalonados entre 0 e 1 com finalidade de melhorar o desempenho da LSTM. Por último, a terceira parte seria a verificação de qual nível de *Dropout* seria ideal para diminuir as métricas de erro da previsão.

Terminando esses estudos, nossa rede está pronta para passar pelo processo iterativo que definira o número de nós mais efetivos em cada camada recorrente.

4 Resultados

4.1 Utilização do recurso "Validação" da LSTM

O banco de dados que foi utilizado já foi processado em modo janela, e utilizando dele, dividimos em 3 partes, onde 80% dos dados serviram para o treinamento da rede neural, 10% para a validação dos dados treinados e 10% para o teste de previsão.

Realizado os testes com e sem a validação do LSTM foram obtidos os resultados abaixo:

Tabela 1 – Métricas da LSTM com validação e sem validação.

Métricas	Sem validação	Com validação
Erro treinamento MSE:	1.14	0.94
Erro treinamento MAPE:	4.12%	3.78%
Erro treinamento MaxError:	4.5	4.28
Erro teste MSE:	2.28	1.82
Erro teste MAPE:	5.64%	4.78%
Erro teste MaxError:	5.45	4.9

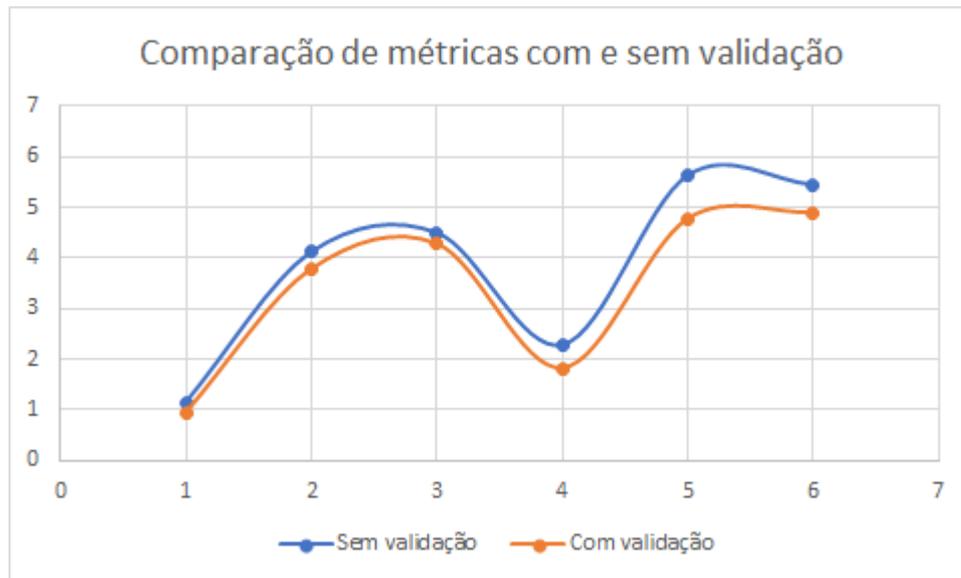
As métricas dispostas na Tabela 1 são o Erro médio quadrático (MSE) citado na subseção 2.3.1, *Mean absolute percentade error* (MAPE) citado na subseção 2.3.2 e Erro máximo citado na subseção 2.3.3, todos sendo avaliados no ambiente de treinamento e posteriormente no ambiente de teste.

Como visto na Tabela 1, todos os valores de métricas dos erros foram menores com a validação. Isso pode se explicado pelo fato de que o conjunto de teste de validação avalia a capacidade da rede neural de prever com base em novas condições que não faziam parte do conjunto de treinamento, fazendo, nesse caso, diminuir os erros.

Analisando as curvas da figura 22 é percebido que a rede neural com o método de validação é mais efetiva pelo seu baixo valor em suas métricas em relação sem validação, vendo que ela compreende uma área menor em relação ao eixo x.

Os números dispostos no eixo x da figura 22 representa as métricas avaliativas da tabela 1, onde começa em 1 sendo o "Erro treinamento MSE" e termina em 6 sendo "erro teste MaxError"

Figura 22 – Comparação de curvas das métricas com e sem validação.



Fonte: Elaborado pelo autor 2022

4.2 Normalização dos dados de entrada

A normalização dos dados consiste em dividir os valores da tabela por uma constante, fazendo com que se encaixem entre 0 e 1, para que facilite o processamento da rede neural em uso.

A equação usada para normalizar os dados 4.1 e em seguida seu inverso 4.2 para "desnormalizar" e serem feitos os cálculos das métricas:

$$data = \frac{(data - datamin)}{datamax - datamin} \quad (4.1)$$

$$data = data * (datamax - datamin) + datamin \quad (4.2)$$

Os resultados para ambos foram avaliados e montados em uma tabela para comparação. Os resultados foram mesclados com a próxima variação e foram referenciados na Tabela 2 e Figura 23.

4.2.1 Variações de *Dropout*

Juntamente com a verificação de entrada dos dados, sendo eles normalizados ou não, foi feito o estudo de caso para variações dos valores de *Dropout*, sendo eles indo de 0 a 0.9, seguindo passos de 0.1. Os valores obtidos foram dispostos nas Figuras 23 e 24 e nas Tabelas 2 e 3 para melhor análise.

4.2.1.1 Variações de *Dropout* com normalização dos dados de entrada

Na Tabela 2 abaixo foram dispostos todos os valores de *Dropout* escolhidos dentro do for iterativo. Todos os valores estão na primeira coluna da tabela. No cabeçalho estão dispostas todas as métricas avaliativas, incluindo as métricas de validação, as quais foram inclusas no trabalho depois de validadas na subseção 4.1.

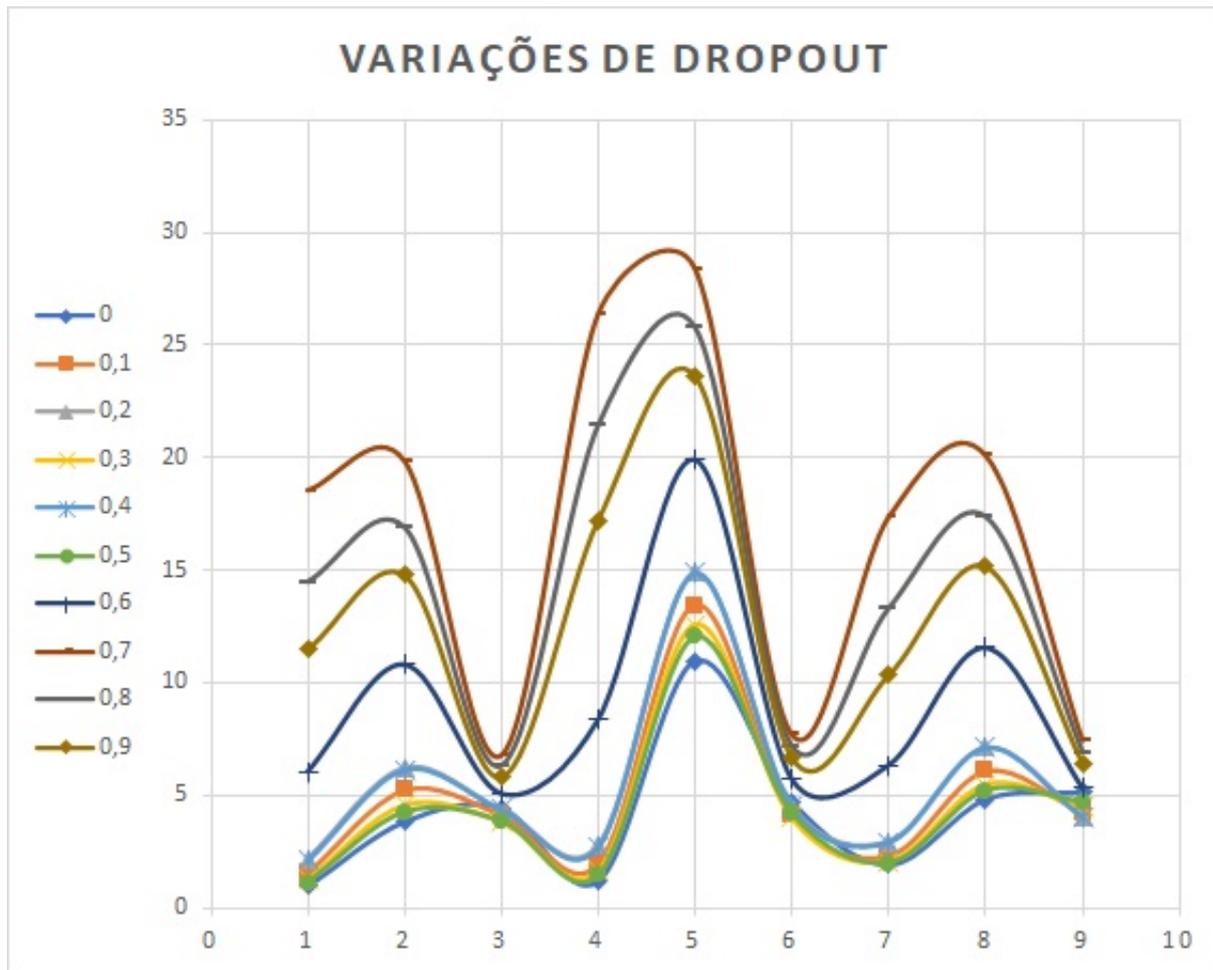
Tabela 2 – Métricas da LSTM com normalização e com seus respectivos valores de *Dropout*.

	treino MSE:	treino MAPE%:	treino MaxError:	val MSE:	val MAPE%:	val MaxError:	teste MSE:	teste MAPE%:	teste MaxError:
0	0,96	3,83	4,44	1,2	10,95	4,71	1,92	4,82	5,13
0,1	1,56	5,22	4,14	1,96	13,43	4,15	2,33	6,07	4,25
0,2	2,15	6,2	4,36	2,76	14,88	4,51	2,92	7,11	4,04
0,3	1,27	4,57	3,83	1,66	12,57	3,99	2,09	5,49	4,5
0,4	2,11	6,16	4,4	2,73	14,9	4,48	2,94	7,15	4,04
0,5	1,12	4,25	3,85	1,43	12,03	4,18	1,99	5,21	4,7
0,6	6,06	10,82	5,07	8,38	19,94	5,7	6,34	11,57	5,32
0,7	18,54	19,84	6,76	26,37	28,38	7,68	17,35	20,08	7,42
0,8	14,48	16,91	6,29	21,46	25,82	7,16	13,31	17,39	6,9
0,9	11,47	14,78	5,79	17,23	23,65	6,65	10,33	15,17	6,39

Analisando a Figura 23, é visto que as 3 linhas mais baixas são as que representam a configuração da rede neural com 0, 0.3 e 0.5 como valores de *Dropout*, e, portanto, representam as melhores configurações pois resultam no menor valor das métricas avaliativas. O eixo horizontal da tabela representa os erros que foram as métricas avaliativas usadas para mensurar a eficiência da configuração. O eixo vertical representa a amplitude das métricas.

Na legenda ao lado esquerdo representa a variação do parâmetro de *Dropout* indo de 0 a 9, que representa o quanto a camada esquecera dos dados de entrada que seguiram para a saída do nó atual. É visto também que as curvas, com os dados normalizados seguem um padrão, desde seu valor menor ao maior, mas perde para o modelo sem normalização, pois todas as suas curvas contém "acidentes" que a deixam com sua área sobre o eixo x maior que as demais da Figura 24.

Outro padrão é observável quando analisamos os pontos em que as curvas se encontram ou ficam muito próximas, e elas são os erros máximos, nos trazendo que existe um limite superior para todas elas em comum, provando a utilidade da função tanh, impedindo os valores de erros estourarem para valores demasiadamente grandes.

Figura 23 – Variações de *Dropout* com normalização

Fonte: Elaborado pelo autor 2022

4.2.1.2 Variações de *Dropout* sem normalização dos dados de entrada

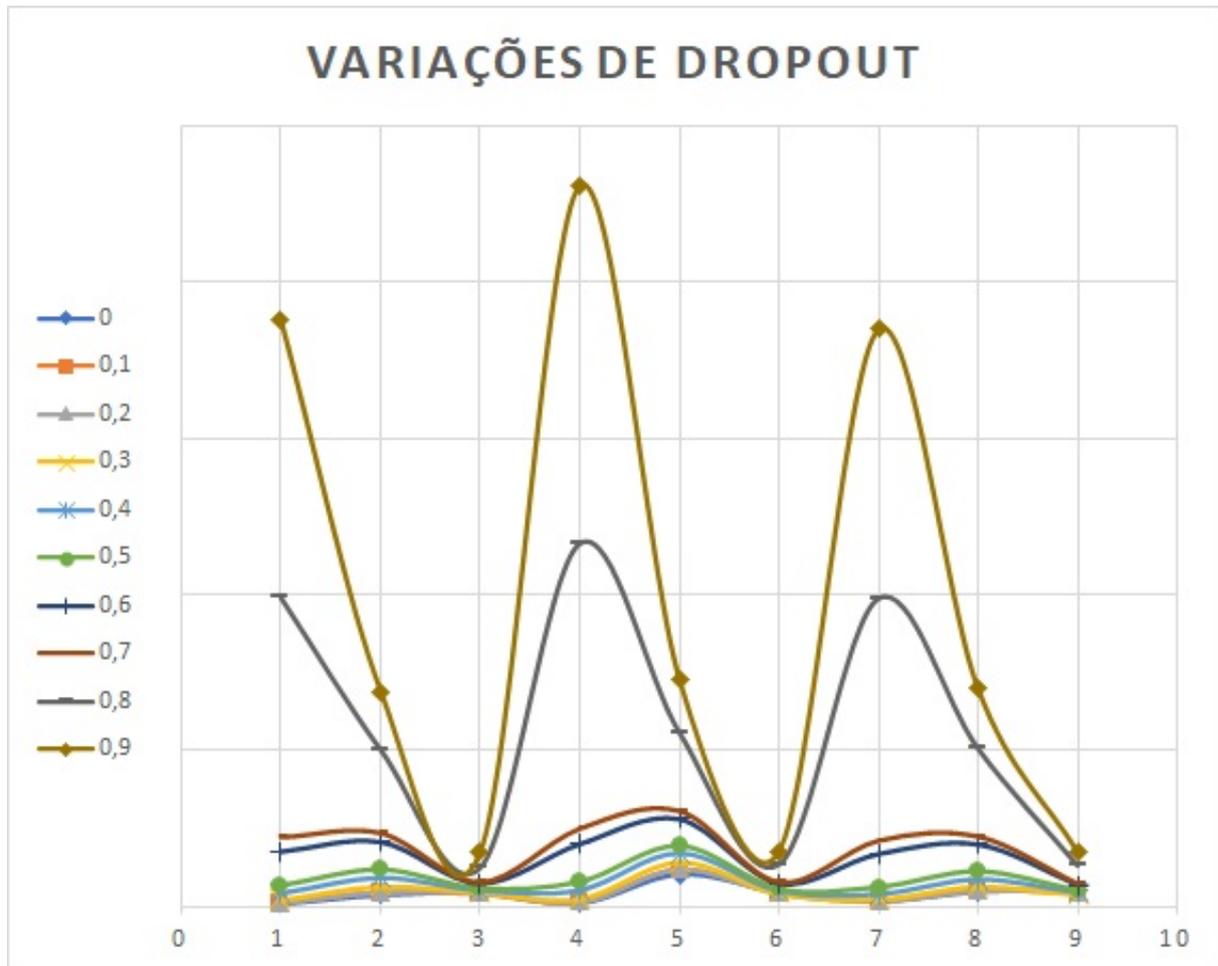
Na Tabela 3 foram dispostos todos os valores de iteração, novamente, só que sem a prévia normalização dos dados. Nas amplitudes das métricas é possível encontrar valores menores em maioria dos valores usados de *Dropout*.

Tabela 3 – Métricas da LSTM sem normalização e seus respectivos valores de *Dropout*.

	treino MSE:	treino MAPE%:	treino MaxError:	val MSE:	val MAPE%:	val MaxError:	teste MSE:	teste MAPE%:	teste MaxError:
0	0,96	3,74	4,4	1,23	10,18	4,84	1,9	4,8	5,1
0,1	1,19	4,52	4,04	1,5	11,63	4,27	1,86	5,04	4,53
0,2	1,29	4,71	4,53	1,5	12,18	4,31	1,97	5,35	4,53
0,3	2,23	6,41	4,6	2,63	13,89	4,23	2,49	6,38	3,86
0,4	4,22	9,13	5,17	5,22	16,98	4,98	3,99	8,68	4,48
0,5	6,62	11,91	5,71	7,88	19,74	5,55	6,1	11,31	5,05
0,6	17,62	20,68	7,34	20,04	28,07	7,37	16,81	19,91	6,88
0,7	22,13	23,4	7,87	24,81	30,41	7,91	20,98	22,31	7,39
0,8	98,99	50,25	12,48	116,45	55,77	13,52	98,82	50,62	13,18
0,9	187,86	68,87	17,28	231,14	72,84	17,47	185	69,66	17,28

A partir dos dados acima na Tabela 3 foi plotado suas curvas relacionadas, é possível avaliar que a partir do momento que os parâmetros de *Dropout* da rede começa a aumentar, seu erros também aumentam, resultando em que a sua melhor configuração seria sem o *Dropout*, ou seja, quando assume o valor 0.

Figura 24 – Variações de *Dropout* sem normalização



Fonte: Elaborado pelo autor 2022

4.3 Teste iterativo de nós para LSTM

Após os testes verificamos que a melhor rede possível, dentre as configurações utilizadas, seria a rede com a validação dos dados de entrada, seguido da sua normalização, com a configuração de *Dropout* entre os valores de 0, 0,3 e 0,5. A partir dessas configurações será feito um *for* para percorrer todas as combinações de nós presentes na rede LSTM utilizada, com passos de 5 em 5. Dos valores de *Dropout* possíveis foram escolhidos o "0" e "0,3" para o teste iterativo e verificar qual das configurações resulta as menores métricas, seu respectivos gráficos serão ilustrados mais a frente nas Figuras 27 e 28 respectivamente.

5 Conclusão

Após as iterações foram encontradas as métricas avaliativas de erros abaixo para dois valores de *Dropout*, onde na Tabela 4 abaixo é mostrado os dados do primeiro exemplo:

Tabela 4 – Métricas da melhor configuração da LSTM com *Dropout* 0.3.

Métricas	Valores
Erro treinamento MSE	2.01
Erro treinamento MAPE	5.92%
Erro treinamento MaxError	3.99
Erro Val MSE	2.72
Erro Val MAPE	13.78%
Erro Val MaxError	4.24
Erro teste MSE	2.41
Erro teste MAPE	6.22%
Erro teste MaxError	3.87

A Figura 25 é uma captura de tela do software Google Colab[®] que mostra a quantidade de nós (neurônios) utilizados em cada camada, seguido de suas ativações, para uma melhor configuração da LSTM com validação, sem normalização dos dados e com o valor de *Dropout* de 0.3.

Figura 25 – Melhor configuração de neurônios na LSTM com *Dropout* 0.3.

```
model = Sequential([layers.Input((4, 1)),
                    layers.LSTM(6),
                    layers.Dropout(0.3),
                    layers.Dense(1, activation='relu'),
                    layers.Dense(21, activation='relu'),
                    layers.Dense(1)])
```

Fonte: Elaborado pelo autor 2022

Tabela 5 – Métricas da melhor configuração da LSTM com *Dropout* 0.

Métricas	Valores
Erro treinamento MSE	1.02
Erro treinamento MAPE	4.01%
Erro treinamento MaxError	4.23
Erro Val MSE	1.27
Erro Val MAPE	10.15%
Erro Val MaxError	4.82
Erro teste MSE	1.92
Erro teste MAPE	4.9%
Erro teste MaxError	5.05

Provando que a configurações com os parâmetros abaixo é a melhor para prever a demanda energética, usando esse banco de dados.

Figura 26 – Melhor configuração de neurônios na LSTM

```

model = Sequential([layers.Input((4, 1)),
                    layers.LSTM(61),
                    layers.Dropout(0),
                    layers.Dense(21, activation='relu'),
                    layers.Dense(26, activation='relu'),
                    layers.Dense(1)])

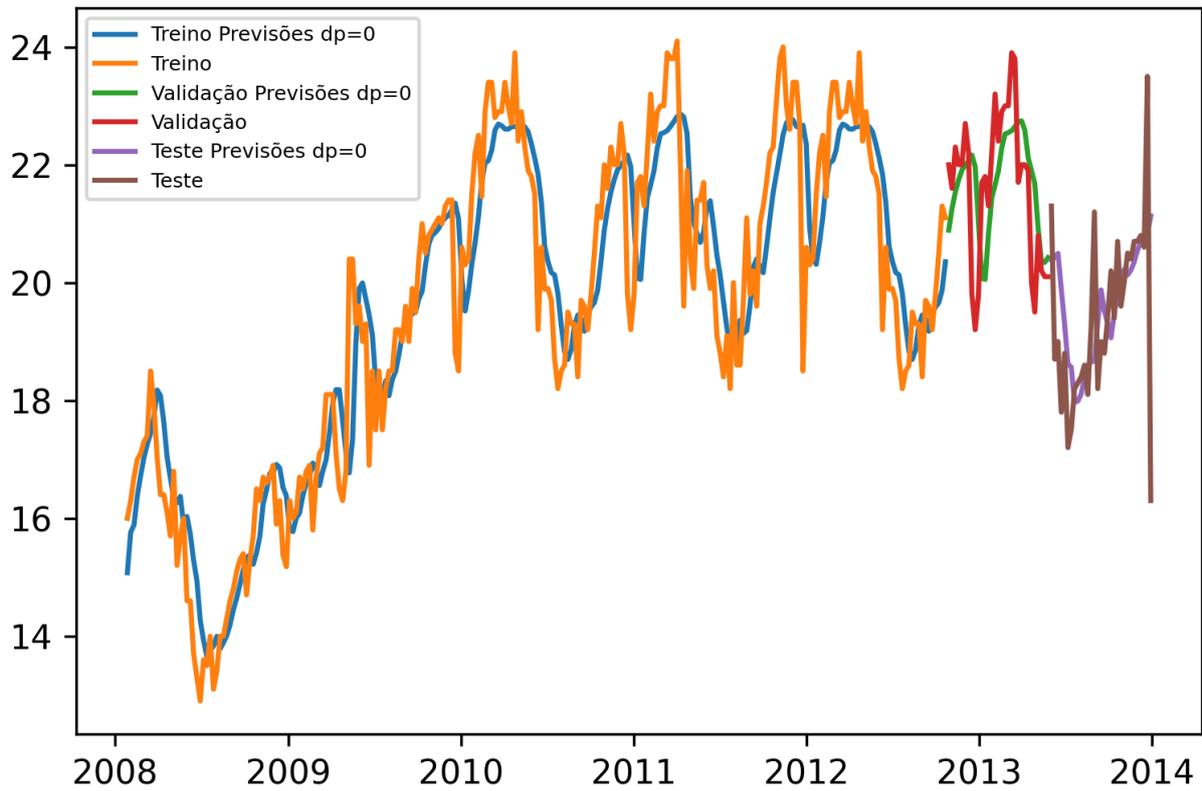
```

Fonte: Elaborado pelo autor 2022

É possível ver que, usando menos neurônios nas camadas, podemos obter o resultado melhor que usando mais, reduzindo o custo computacional para chegar no mesmo objetivo. A normalização dos dados não foi de boa utilidade nesse banco de dados. O seu uso causou curvas maiores como podemos ver no gráfico 23. A validação, ferramenta do LSTM, ajudou na diminuição das métricas avaliativas. O *Dropout* não resultou em melhoria como foi previsto, neste banco de dados utilizado. Mostrando as diferenças de métricas avaliativas podemos comparar as duas configurações de redes neurais realizando as previsões de dados.

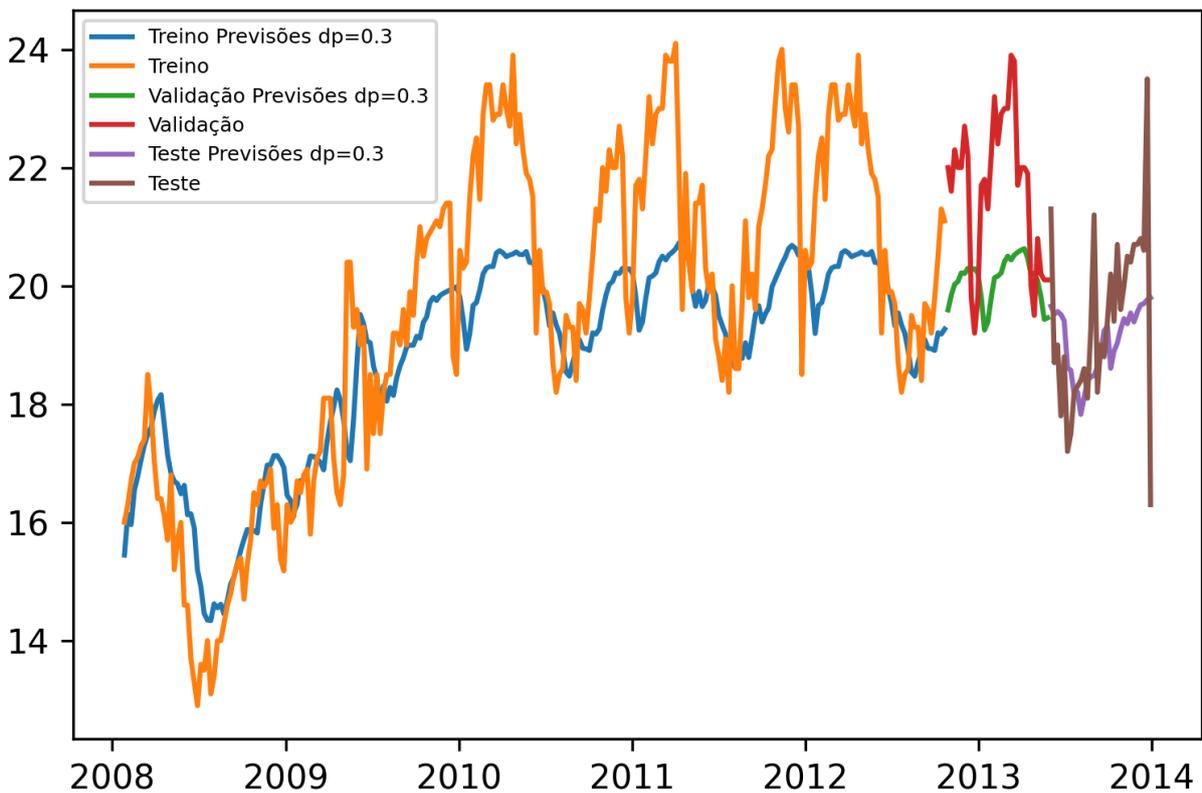
Na Figura 27 (a que usa o $dp=0.3$) é possível ver como a curva de predição acompanha os dados de treino mas não está tão próxima como as da Figura 28 (que usa $dp=0$)

Figura 27 – Gráfico de previsões usando $dp=0$



Fonte: Elaborado pelo autor 2022

Figura 28 – Gráfico de previsões usando $dp=0.3$



Fonte: Elaborado pelo autor 2022

6 Trabalhos futuros

Após conclusões dos dados tomados nesse trabalho, foi possível prever alguns trabalhos que poderiam dar continuidade a esse, ou melhorar os resultados obtidos no mesmo. O primeiro seria o aumento do *Learning Rate*, função interna do LSTM, o otimizador Adam, que foi definido nesse trabalho como 0,001 por convenção e que poderia ser utilizado um *for* iterativo podendo se obter melhores resultados. Outro seria o aumento ou diminuição de *epochs*, e analisar se as métricas poderiam aumentar ou diminuir, com o objetivo de diminuir o processamento ou aumentar a acuracidade das métricas. A terceira ideia de trabalho seria a comparação dos resultados deste trabalho com o desempenho de outras redes neurais, utilizando o mesmo banco de dados, com a mesma periodicidade, para provar a eficiência da rede neural em frente as outras.

Referências

- ABRADEE. Visão geral do setor - abra-dee. abra-dee. disponível em: <<https://www.abra-dee.org.br/setor-eletrico/visao-geral-do-setor/>>. acesso em: 14 nov. 2022. In: . [S.l.: s.n.], 2018. Citado na página 19.
- ALMESHAI EI, E.; SOLTAN, H. A methodology for electric power load forecasting. *Alexandria Engineering Journal*, Elsevier, v. 50, n. 2, p. 137–144, 2011. Citado na página 15.
- AN, N. et al. Using multi-output feedforward neural network with empirical mode decomposition based signal filtering for electricity demand forecasting. *Energy*, Elsevier, v. 49, p. 279–288, 2013. Citado na página 14.
- ARMSTRONG, J. S.; COLLOPY, F. Error measures for generalizing about forecasting methods: Empirical comparisons. *International journal of forecasting*, Elsevier, v. 8, n. 1, p. 69–80, 1992. Citado na página 29.
- BROWNLEE, J. A gentle introduction to the rectified linear unit (relu) - machinelearningmastery.com. machinelearningmastery.com. disponível em: <<https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/>>. acesso em: 22 nov. 2022. 2019. Citado na página 38.
- CHEN, K.-Y.; WANG, C.-H. A hybrid sarima and support vector machines in forecasting the production values of the machinery industry in taiwan. *Expert Systems with Applications*, Elsevier, v. 32, n. 1, p. 254–264, 2007. Citado na página 15.
- COLAH. Understanding lstm networks – colahs blog. github.io. disponível em: <<https://colah.github.io/posts/2015-08-understanding-lstms/>>. acesso em: 18 nov. 2022. Github, 2015. Citado 6 vezes nas páginas 23, 24, 25, 26, 27 e 28.
- EPE. Matriz energética. epe.gov.br. disponível em: <<https://www.epe.gov.br/pt/abcedenergia/matriz-energetica-e-eletrica>>. acesso em: 14 nov. 2022. In: . [S.l.: s.n.], 2018. Citado 2 vezes nas páginas 19 e 20.
- GARCÍA-ASCANIO, C.; MATÉ, C. Electric power demand forecasting using interval time series: A comparison between var and implp. *Energy Policy*, Elsevier, v. 38, n. 2, p. 715–725, 2010. Citado na página 15.
- HINTON, G. E.; NOWLAN, S. Simplifying neural networks by soft weight sharing. In: ADDISON-WESLEY PUBLISHING CO. *SANTA FE INSTITUTE STUDIES IN THE SCIENCES OF COMPLEXITY-PROCEEDINGS VOLUME-*. [S.l.], 1994. v. 20, p. 373–373. Citado na página 31.
- HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. *Neural computation*, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado 3 vezes nas páginas 16, 22 e 23.
- HOR, C.-L.; WATSON, S. J.; MAJITHIA, S. Daily load forecasting and maximum demand estimation using arima and garch. In: IEEE. *2006 International Conference on Probabilistic Methods Applied to Power Systems*. [S.l.], 2006. p. 1–6. Citado na página 15.

KAM, K. M. *Stationary and non-stationary time series prediction using state space model and pattern-based approach*. [S.l.]: The University of Texas at Arlington, 2014. Citado na página 15.

KAVAKLIOGLU, K. et al. Modeling and prediction of turkeys electricity consumption using artificial neural networks. *Energy Conversion and Management*, Elsevier, v. 50, n. 11, p. 2719–2727, 2009. Citado na página 15.

KHASHEI, M.; BIJARI, M. A novel hybridization of artificial neural networks and arima models for time series forecasting. *Applied soft computing*, Elsevier, v. 11, n. 2, p. 2664–2675, 2011. Citado na página 15.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. Citado na página 38.

KO, C.-N.; LEE, C.-M. Short-term load forecasting using svr (support vector regression)-based radial basis function neural network with dual extended kalman filter. *Energy*, Elsevier, v. 49, p. 413–422, 2013. Citado na página 15.

LETSCHERT, V. The boom of electricity demand in the residential sector in the developing world and the potential for energy efficiency. 2008. Citado na página 14.

LI, Y.; WANG, J. The load forecasting model based on bayes-grnn. *J. Softw.*, v. 7, n. 6, p. 1273–1280, 2012. Citado na página 15.

LINEESH, M.; MINU, K.; JOHN, C. J. Analysis of nonstationary nonlinear economic time series of gold price: A comparative study. In: *International mathematical forum*. [S.l.: s.n.], 2010. v. 5, n. 34, p. 1673–1683. Citado na página 15.

MEDEIROS, R. Á. O. d. et al. Previsao de demanda no medio prazo utilizando redes neurais artificiais em sistemas de distribuicao de energia eletrica. Universidade Federal da Paraíba, 2016. Citado 2 vezes nas páginas 21 e 37.

NETO, A. B. F.; CORRÊA, W. L. R.; PEROBELLI, F. S. Consumo de energia e crescimento econômico: uma análise do brasil no período 1970-2009. *Análise Econômica*, v. 34, n. 65, 2016. Citado na página 14.

PANIGRAHI, S.; KARALI, Y.; BEHERA, H. Time series forecasting using evolutionary neural network. *International Journal of Computer Applications*, Foundation of Computer Science, v. 75, n. 10, 2013. Citado na página 15.

PISHRO-NIK, H. Introduction to probability, statistics, and random processes. 2016. Citado na página 29.

ROBBA KAGAN, O. *KAGAN, Nelson e OLIVEIRA, Carlos César Barioni de e ROBBA, Ernesto João. Introdução aos sistemas de distribuição de energia elétrica. . São Paulo: Edgard Blucher. . Acesso em: 13 nov. 2022. , 2005*. [S.l.]: Edgard Blucher, 2005. Citado 2 vezes nas páginas 20 e 21.

SAK, H.; SENIOR, A. W.; BEAUFAYS, F. Long short-term memory recurrent neural network architectures for large scale acoustic modeling. 2014. Citado na página 16.

SRIVASTAVA, N. et al. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, JMLR. org, v. 15, n. 1, p. 1929–1958, 2014. Citado 4 vezes nas páginas 30, 31, 32 e 33.

SVERMA, Y. A complete understanding of dense layers in neural networks. analytics india magazine. disponível em: <<https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/>>. acesso em: 22 nov. 2022. 2022. Citado na página 38.

XIONG, H. Y.; BARASH, Y.; FREY, B. J. Bayesian prediction of tissue-regulated splicing using rna sequence and cellular context. *Bioinformatics*, Oxford University Press, v. 27, n. 18, p. 2554–2562, 2011. Citado na página 31.

YAO, T. Zero-carbon energy kyoto 2012. In: SPRINGER. *Special Edition of the Joint Symposium Energy Science in the Age of Global Warming of the Kyoto University Global COE Program and the JGSEE/CEE-KMUTT*. [S.l.], 2013. Citado na página 14.

ZHANG, G. P. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, Elsevier, v. 50, p. 159–175, 2003. Citado na página 15.

ZHENG, J. et al. Electric load forecasting in smart grids using long-short-term-memory based recurrent neural network. In: IEEE. *2017 51st Annual conference on information sciences and systems (CISS)*. [S.l.], 2017. p. 1–6. Citado na página 15.