

Silvio Silva dos Santos

# **Previsão da Produção de uma Usina Fotovoltaica Usando Redes Neurais Artificiais**

Goiânia

2019



---

**TERMO DE CIÊNCIA E DE AUTORIZAÇÃO PARA DISPONIBILIZAR  
VERSÕES ELETRÔNICAS DE TRABALHO DE CONCLUSÃO DE CURSO DE  
GRADUAÇÃO NO REPOSITÓRIO INSTITUCIONAL DA UFG**

---

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio do Repositório Institucional (RI/UFG), regulamentado pela Resolução CEPEC nº 1204/2014, sem ressarcimento dos direitos autorais, de acordo com a Lei nº 9610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou *download*, a título de divulgação da produção científica brasileira, a partir desta data.

**1. Identificação do Trabalho de Conclusão de Curso de Graduação (TCCG):**

Nome completo do autor: Silvio Silva dos Santos

Título do trabalho: Previsão da Produção de uma Usina Fotovoltaica Usando Redes Neurais Artificiais

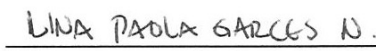
**2. Informações de acesso ao documento:**

Concorda com a liberação total do documento ☒ SIM ☐ NÃO<sup>1</sup>

Havendo concordância com a disponibilização eletrônica, torna-se imprescindível o envio do(s) arquivo(s) em formato digital PDF do TCCG.

  
(Nome completo do autor)<sup>2</sup>

Ciente e de acordo:

  
(Nome completo do orientador)<sup>2</sup>

Data: 13 / 12 / 2019

---

<sup>1</sup> Neste caso o documento será embargado por até um ano a partir da data de defesa. A extensão deste prazo suscita justificativa junto à coordenação do curso. Os dados do documento não serão disponibilizados durante o período de embargo.

Casos de embargo:

- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

<sup>2</sup> As assinaturas devem ser originais sendo assinadas no próprio documento, imagens coladas não serão aceitas.

Silvio Silva dos Santos

# **Previsão da Produção de uma Usina Fotovoltaica Usando Redes Neurais Artificiais**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina TCC, do curso de Engenharia Elétrica da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Engenheiro Eletricista.

Universidade Federal de Goiás – UFG

Escola de Engenharia Elétrica, Mecânica e de Computação

Trabalho de Conclusão de Curso

Orientador: Lina Paola Garces Negrete

Coorientador: Ricardo Henrique Fonseca Alves

Goiânia

2019

Silvio Silva dos Santos

Previsão da Produção de uma Usina Fotovoltaica Usando Redes Neurais Artificiais/  
Silvio Silva dos Santos. – Goiânia, 2019-  
63f.

Orientador: Lina Paola Garces Negrete

Trabalho de Conclusão de Curso) – Universidade Federal de Goiás – UFG  
Escola de Engenharia Elétrica, Mecânica e de Computação  
Trabalho de Conclusão de Curso, 2019.

1. Previsão de geração fotovoltaica. 2. Long Short-Term Memory. 3. Redes neurais recorrente. 4. Técnica da janela. I. Orientador: Profa. Dra. Lina Paola Garces Negrete. II. Universidade Federal de Goiás. III. Escola de Engenharia Elétrica, Mecânica e da Computação. IV. Título: Previsão da Produção de uma Usina Fotovoltaica Usando Redes Neurais Artificiais.

Silvio Silva dos Santos

## **Previsão da Produção de uma Usina Fotovoltaica Usando Redes Neurais Artificiais**

Trabalho de Conclusão de Curso de Graduação, apresentado à disciplina TCC, do curso de Engenharia Elétrica da Universidade Federal de Goiás, como requisito parcial para obtenção do título de Engenheiro Eletricista.

Trabalho aprovado. Goiânia, 05 de novembro de 2019:

LINA PAOLA GARCES N.

Lina Paola Garces Negrete  
Orientador

Gelson Brito

Professor  
Convidado 1

Leuz J.R. Jr.

Convidado 2

Goiânia  
2019

*À minha família,  
que com muito amor me fizeram chegar aqui.*





# Agradecimentos

Agradeço aos meus pais, que me criaram com muito amor enfrentando os mais duros desafios sempre de cabeça erguida, sem abalar, com integridade e muito trabalho duro. Agradeço aos meus tios que também deram muito suporte quando meus pais não podiam, assim como meu irmão e minhas irmãs.

Agradeço a minha orientadora Profa. Dra. Lina , e ao meu coorientador M<sup>e</sup>. Ricardo Henrique Fonseca Alves. Por terem me dado a honra e oportunidade de trabalhar com eles. Agradeço, principalmente, a professora Lina por toda ajuda, apoio e tempo disponibilizado durante o período de desenvolvimento deste trabalho. Agradeço também à Profa. Dra. Cacilda de Jesus Ribeiro, que me ajudou e apoiou em muitos momentos durante o curso. À todos os professores que tive ao longo do curso. Aos meus amigos e colegas, que sempre me incentivaram e não me deixaram desistir. Em especial, a Guilherme Ayres Himmen, pela amizade e companheirismo ao longo do curso. À todas as pessoas que convivem ou conviveram comigo nos últimos meses, e que me ajudaram de alguma forma, direta ou indiretamente, na realização do presente trabalho. Agradeço também, a Universidade Federal de Goiás - Campus Colemar Natal e Silva.



*Love is wise; hatred is foolish. In this world,  
which is getting more and more closely interconnected,  
we have to learn to tolerate each other, we have to learn to put up  
with the fact that some people say things that we don't like. We can only  
live together in that way. But if we are to live together, and not die together,  
we must learn a kind of charity and a kind of tolerance,  
which is absolutely vital to the continuation of human  
life on this planet.  
(Bertrand Arthur William Russell)*



# Resumo

A previsão de geração de energia solar, ou fotovoltaica, é tema de muita complexidade, pois envolve o desenvolvimento de modelos estatísticos complexos. O presente trabalho procura contornar a complexidade dos modelos estatísticos, através do uso de um tipo rede neural recorrente, trazendo assim, uma dinâmica diferente para o problema. Empregou-se no presente estudo uma rede *Long Short-Term Memory* (LSTM) para realizar a previsão em horizontes de 15 minutos, uma hora e um dia a frente. Utilizou-se a técnica de janelamento junto com a rede LSTM para fazer a previsão, e evitar, dessa forma, o uso de regressão linear visto a natureza estocástica do problema abordado. Os dados de geração de energia e climáticos foram coletados no intervalo de 6:00 às 18:00, da usina fotovoltaica e estação meteorológica, respectivamente instaladas no bloco B da Escola de Engenharia Elétrica, Mecânica e de Computação (EMC) da Universidade Federal de Goiás (UFG). Foram implementados três modelos diferentes de redes neurais LSTM e treinadas com os dados dos três horizontes de tempo (15 minutos, uma hora e um dia a frente), e assim, selecionando os modelos que apresentaram o menor erro, entre o valor real e o valor previsto.

**Palavras-chave:** Previsão de geração fotovoltaica. Redes neurais recorrentes. *Long Short-Term Memory* - *LSTM*. Técnica da janela.



# Abstract

The solar or photovoltaic power generation forecasting is a very complex subject, since it involves the development of complex statistical models. The present work seeks to circumvent the complexity of statistical models through the use of a recurrent neural network type, thus bringing a different dynamic to the problem. Long Short-Term Memory-LSTM networks were used to forecast at horizons of 15 minutes, one hour and one day ahead. The windowing technique was used together with the LSTM network to make the prediction, thus avoiding the use of linear regression given the stochastic nature of the problem addressed. Power generation and climate data were collected from 6:00 to 18:00 from the photovoltaic plant and weather station, respectively installed at the B block of the School of Electrical, Mechanical and Computer Engineering (EMC) of the Federal University of Goiás (UFG). Three different LSTM neural network models were implemented and trained with the data from the three time horizons (15 minutes ahead, one hour ahead and one day ahead), thus selecting the models that presented the smallest error, between the real value and the predicted value.

**Keywords:** Photovoltaic generation forecasting. Recurrent neural networks. Long Short-Term Memory - LSTM. Window technique.





# Lista de ilustrações

Figura 2.1 – Evolução da potência instalada – micro e mini-geração. . . . .	22
Figura 2.2 – Passado, presente e futuro da energia gerada em casa. . . . .	22
Figura 2.3 – Estrutura de um neurônio. . . . .	26
Figura 2.4 – Modelo de neurônio artificial. . . . .	27
Figura 2.5 – Rede neural artificial. . . . .	29
Figura 2.6 – Função ReLU. . . . .	31
Figura 2.7 – Função sigmoide logística. . . . .	31
Figura 2.8 – Uma rede neural é parametrizada pelos seus pesos. . . . .	32
Figura 2.9 – A função de perda mede a qualidade da saída da rede. . . . .	33
Figura 2.10 – A perda é usada como sinal de realimentação para ajustar os pesos. . .	33
Figura 2.11 – Célula de memória LSTM. . . . .	35
Figura 2.12 – Estado da célula de memória LSTM. . . . .	36
Figura 2.13 – <i>Gate</i> genérico. . . . .	36
Figura 2.14 – Atuação do <i>forget gate</i> . . . . .	37
Figura 2.15 – Atuação do <i>input gate</i> . . . . .	37
Figura 2.16 – Atualização do estado da célula. . . . .	38
Figura 2.17 – Atuação do <i>output gate</i> . . . . .	38
Figura 3.1 – Site da estação meteorológica localizada na EMC. . . . .	41
Figura 3.2 – Planilha de dados meteorológicos março de 2019. . . . .	42
Figura 3.3 – Site do sistema de monitoramento conectado a rede. . . . .	42
Figura 3.4 – Dados de entrada para médias de 15 minutos. . . . .	43
Figura 3.5 – Teste de correlação <i>data2</i> . . . . .	44
Figura 3.6 – Teste de correlação, conjunto <i>data1</i> . . . . .	44
Figura 3.7 – Teste de correlação, conjunto <i>data3</i> . . . . .	45
Figura 4.1 – Gráfico <i>model3</i> , previsão de 96 pontos . . . . .	53
Figura 4.2 – Gráfico <i>model3</i> , previsão de 36 pontos . . . . .	53
Figura 4.3 – Gráfico <i>model1</i> , previsão de 36 pontos . . . . .	54
Figura A.1 – Principais características estatísticas do conjunto <i>data1</i> . . . . .	61
Figura A.2 – Principais características estatísticas do conjunto <i>data2</i> . . . . .	61
Figura A.3 – Principais características estatísticas do conjunto <i>data3</i> . . . . .	62

# Lista de tabelas

Tabela 4.1 – Valores de erros em kWh, conjunto <i>data1</i> . . . . .	51
Tabela 4.2 – Valores de erros em kWh, conjunto <i>data2</i> . . . . .	52
Tabela 4.3 – Valores de erros em kWh, conjunto <i>data3</i> . . . . .	52
Tabela B.1 – Tempos de treinamento dos modelos com melhores resultados. . . . .	63

# Sumário

<b>1</b>	<b>INTRODUÇÃO</b>	<b>19</b>
<b>1.1</b>	<b>Problema e justificativa</b>	<b>19</b>
<b>1.2</b>	<b>Objetivos principais e específicos</b>	<b>19</b>
<b>1.3</b>	<b>Contribuições</b>	<b>20</b>
<b>1.4</b>	<b>Estrutura do trabalho</b>	<b>20</b>
<b>2</b>	<b>LEVANTAMENTO BIBLIOGRÁFICO</b>	<b>21</b>
<b>2.1</b>	<b>Energia solar</b>	<b>21</b>
<b>2.2</b>	<b>Geração distribuída</b>	<b>21</b>
<b>2.3</b>	<b>Previsão de geração solar</b>	<b>22</b>
<b>2.4</b>	<b>Redes neurais</b>	<b>25</b>
2.4.1	Neurônio biológico	25
2.4.2	Neurônio artificial	26
2.4.3	Entendendo como uma rede neural artificial funciona	28
2.4.4	Arquitetura	29
2.4.5	Processo de treinamento	30
2.4.6	Função de ativação	30
2.4.7	Ciclo de treinamento	32
<b>2.5</b>	<b>Memória de longo prazo</b>	<b>34</b>
2.5.1	Célula de memória LSTM	34
2.5.2	Entendendo a célula de memória LSTM	35
<b>3</b>	<b>METODOLOGIA PARA PREVISÃO DE PRODUÇÃO DE USINA</b>	
	<b>PV</b>	<b>41</b>
<b>3.1</b>	<b>Manipulação dos dados</b>	<b>41</b>
3.1.1	Coleta	41
3.1.2	Depuramento dos dados	43
<b>3.2</b>	<b>Estrutura da RNN</b>	<b>45</b>
3.2.1	Técnica da janela	45
3.2.2	Otimizador e função de perda ( <i>loss function</i> )	47
3.2.3	Cálculo de erro	48
3.2.4	Normalização dos dados	48
3.2.5	Época de treinamento	49
3.2.6	Tamanho do lote <i>Batch size</i>	49
3.2.7	Modelo adotado	49
3.2.8	Programação	50

4	TESTES E ANÁLISE DE RESULTADOS . . . . .	51
4.1	Conjunto de dados <i>data1</i> (15 minutos) . . . . .	51
4.2	Conjunto de dados <i>data2</i> (60 minutos) . . . . .	51
4.3	Conjunto de dados <i>data3</i> (720 minutos) . . . . .	52
4.4	Resultados gráficos . . . . .	52
5	CONCLUSÃO . . . . .	55
5.1	Trabalhos futuros . . . . .	55
	REFERÊNCIAS . . . . .	57
	ANEXOS	59
	ANEXO A – PRINCIPAIS INFORMAÇÕES ESTATÍSTICAS DOS CONJUNTOS DE DADOS TRABALHADOS . . . . .	61
	ANEXO B – TEMPOS DE TREINAMENTO . . . . .	63

# 1 Introdução

Neste trabalho é apresentada uma metodologia para prever a geração fotovoltaica de energia, utilizando ferramentas computacionais, dado um conjunto de várias climáticas. Como teste foi realizada a previsão de produção de energia elétrica de uma usina solar fotovoltaica instalada no bloco B da Escola de Engenharia Elétrica, Mecânica e da Computação da UFG.

## 1.1 Problema e justificativa

O grande crescimento das fontes de energias renováveis na matriz energética brasileira, acaba provocando uma necessidade, a de ferramentas computacionais robustas e eficientes que auxiliem às concessionárias de energia no planejamento da operação a curto prazo. Desta forma, este projeto visa propor uma ferramenta computacional para predição da produção de energia fotovoltaica por meio de dados meteorológicos para auxiliar na gestão do planejamento e operação de redes de distribuição de forma a melhorar a confiabilidade do sistema de energia.

## 1.2 Objetivos principais e específicos

O objetivo principal deste projeto é criar uma ferramenta computacional capaz de prever, de forma satisfatória, a produção de energia de uma usina fotovoltaica utilizando redes neurais artificiais. Os objetivos específicos deste projeto são:

- Realizar como estudo de caso, a previsão de produção da usina implantada na Escola de Engenharia Elétrica, Mecânica e de Computação (EMC) da Universidade Federal de Goiás (UFG). Criar uma ferramenta computacional capaz de prever a produção diária de energia fotovoltaica da usina instalada no Bloco B da EMC;
- Coletar informações meteorológicas e de produção fotovoltaica do sistema implantado no Bloco B da EMC na UFG dos últimos 6 meses;
- Analisar a sensibilidade da rede neural para os diferentes parâmetros de entrada;
- Estruturar uma arquitetura de Rede Neural Recorrente;
- Capacitar o aluno de Graduação acerca dos conceitos e princípios de redes neurais e sistemas fotovoltaicos;

## 1.3 Contribuições

As principais contribuições deste trabalho podem ser resumidas como segue:

- Proposta de um método de previsão de geração fotovoltaica baseado em redes neurais recorrentes, modelo *Log-Short Term Memory - LSTM*. Para isso são usados três conjuntos de dados com amostras tomadas respectivamente a cada 15 minutos, a cada hora e a cada dia;
- Mostra a viabilidade do uso de redes neurais artificiais na previsão de geração fotovoltaica de energia. Contribuindo assim, com estudos posteriores do tema e desenvolvimento de ferramentas computacionais mais robustas que auxiliem na previsão de geração fotovoltaica de energia.

## 1.4 Estrutura do trabalho

Este trabalho é dividido em cinco capítulos, contando com a introdução. Os outros capítulos são:

- Capítulo 2 Levantamento Bibliográfico: neste capítulo faz-se uma breve abordagem sobre energia solar, geração distribuída, previsão de geração solar, redes neurais (neurônio biológico e artificial, arquitetura, treinamento, função de ativação), memória de longo prazo;
- Capítulo 3 Metodologia para previsão de produção de usina PV: neste capítulo introduz-se a metodologia empregada para manipulação dos dados, coleta e depuração, a estrutura da rede neural, técnica da janela e os modelos adotados;
- Capítulo 4 Testes e resultados: neste capítulo exibe-se os resultados obtidos do treinamento dos modelos adotados anteriormente para cada conjunto de dados, comparando-os a partir dos valores de erros;
- Capítulo 5 Conclusão: aqui apresenta-se a conclusão e propostas para trabalhos futuros.

## 2 Levantamento Bibliográfico

### 2.1 Energia solar

O Sol é a principal fonte de energia para a Terra. Além de ser responsável pela manutenção da vida no Planeta, a radiação solar constitui-se numa inesgotável fonte energética, havendo um enorme potencial de sua utilização por meio de sistemas de captação e conversão em outra forma de energia.

Aproveitar a energia gerada pelo Sol é uma das alternativas energéticas mais promissoras para prover a necessária ao desenvolvimento humano. A energia solar fotovoltaica é energia obtida através da conversão direta da luz em eletricidade (Efeito Fotovoltaico), sendo a célula fotovoltaica, o principal dispositivo para esse fim.(CEPEL-CRESEB, 2014).

### 2.2 Geração distribuída

Segundo a ANEEL (2016), a geração distribuída é caracterizada pela instalação de geradores de pequeno porte, normalmente a partir de fontes renováveis ou mesmo utilizando combustíveis fósseis, localizados próximos aos centros de consumo de energia elétrica.

De forma geral, segundo a ANEEL (2016), a presença de pequenos geradores próximos às cargas pode proporcionar diversos benefícios para o sistema elétrico, dentre os quais se destacam a postergação de investimentos em expansão nos sistemas de distribuição e transmissão; o baixo impacto ambiental; a melhoria do nível de tensão da rede no período de carga pesada e a diversificação da matriz energética.

Por outro lado, segundo ANEEL (2016), há algumas desvantagens associadas ao aumento da quantidade de pequenos geradores espalhados na rede de distribuição, tais como: o aumento da complexidade de operação da rede, a dificuldade na cobrança pelo uso do sistema elétrico, a eventual incidência de tributos e a necessidade de alteração dos procedimentos das distribuidoras para operar, controlar e proteger suas redes.

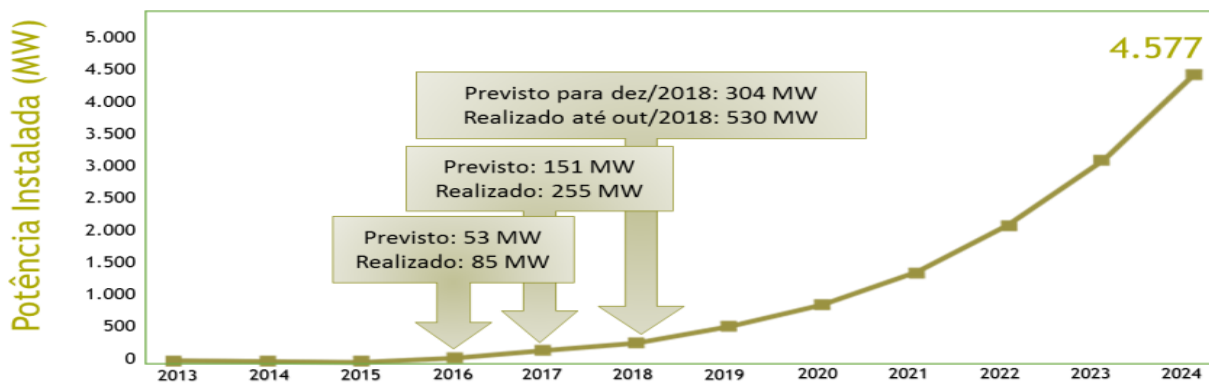
Segundo Nascimento (2017), o Brasil, conforme Ministério de Minas e Energia (2017), possuía, ao final de 2016, 81 MW de energia solar fotovoltaica instalados, sendo 24 MWp de geração centralizada e 57 MWp de geração distribuída.

Desde 17 de abril de 2012, quando a ANEEL criou o Sistema de Compensação de Energia Elétrica, o consumidor brasileiro pode gerar sua própria energia elétrica a partir de fontes renováveis ou cogeração qualificada e inclusive fornecer o excedente para a rede

de distribuição de sua localidade.

Segundo a ANEEL (2018), em termos de potência instalada, a evolução da micro e mini-geração tem se dado em patamares superiores aos projetados pela ANEEL em suas projeções mais otimistas. Verifica-se, da Figura 2.1, que os 500 MW de potência instalada

Figura 2.1 – Evolução da potência instalada – micro e mini-geração.

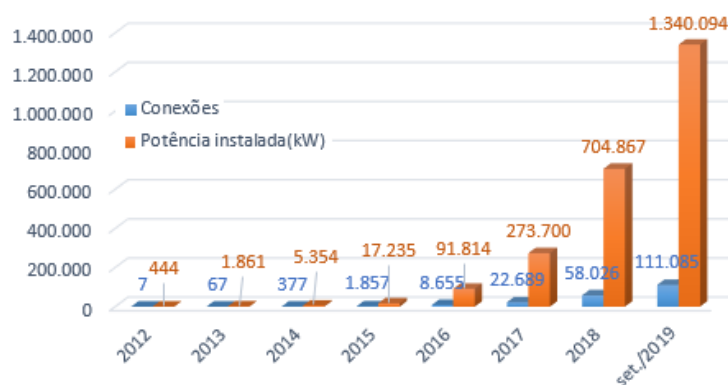


Fonte: ANEEL(ANEEL, 2018).

esperados para o final de 2019 foram atingidos mais de um ano antes da data esperada.

Conforme apresentado na Figura 2.2, que mostra a evolução da potência instalada e do número de consumidores com micro ou mini-geração distribuída até setembro de 2019. Observa-se o crescimento crescente das grandezas, corroborando as previsões do gráfico anterior(Figura 2.1).

Figura 2.2 – Passado, presente e futuro da energia gerada em casa.



Fonte: Adaptado(ANEEL, 2019).

## 2.3 Previsão de geração solar

Prever a produção de energia, especialmente a energia solar, é muito importante para o planejamento do sistema, pois permite que várias ações de gestão da energia sejam



realizadas. Neste item serão apresentados trabalhos que abordam a questão de previsão da produção de energia solar no curtíssimo prazo.

Segundo Lopes (2010), ao contrário do que acontece para a previsão de vento, onde os modelos de previsão se encontram num ponto relativamente maduro, na previsão de irradiação solar os modelos e métodos existentes estão ainda numa fase muito embrionária. Os modelos mais comuns envolvem imagens de satélite e dados meteorológicos para fazerem a previsão de energia solar.

Ainda segundo Lopes (2010), na literatura são encontrados quatro horizontes temporais para previsão da radiação solar.

- Previsão a muito curto prazo: É uma previsão onde o horizonte temporal pode variar de alguns segundos até uma hora. Este tipo de previsões é feito com o auxílio de dados recolhidos no local para o qual se pretende fazer a previsão;
- Previsão a curto prazo: Tipo de previsão com horizonte temporal compreendido entre 1 hora e 7 dias;
- Previsão a longo prazo: Tipo de previsão com horizonte temporal compreendido entre 7 dias e vários anos. Este é um tipo de previsão que é baseada em simulação do recurso energético;
- Previsão a muito longo prazo: Tipo de previsão com horizonte temporal para várias dezenas de anos. Este tipo de previsões do recurso solar, é baseada nos estudos sobre as alterações climáticas.

No trabalho de Santos (2010), é desenvolvida uma abordagem completamente diferente em duas fases distintas, onde em uma fase foi usada uma RNA, para descrever a evolução temporal das nuvens ao longo do tempo. Assim, na primeira fase foi prevista a radiação global, bem como, a radiação solar no local onde o sistema de aquisição de imagens estava instalado. Na segunda fase da metodologia, é proposto um modelo, baseado em RNA (Rede Neural Artificial), que tem por objectivo converter os valores de radiação solar, previstas na primeira fase, num valor adequado de potência fornecida por uma instalação fotovoltaica a cada 5 minutos, tendo em conta a influência de outros fatores climáticos. O autor conclui que obteve-se bons resultados, com a modelagem. Como teve a oportunidade de analisar, a relação entre a produção da central em estudo e a radiação solar não é meramente linear. A utilização de RNA, nesta fase da metodologia, permitiu evitar a utilização de modelos paramétricos, que são dependentes da qualidade dos parâmetros fornecidos pelos fabricantes. A utilização de inteligência artificial para o cálculo da produção da central só foi possível devido à existência de dados históricos de geração em diferentes condições meteorológicas de operação do gerador.

Segundo Andrade (2010) citando Papalexopoulos e Hesterber (1990) que descrevem a previsão de demanda de energia elétrica no curto prazo como necessária para determinar os custos e efeitos da capacidade estimada e de trocas de energia. Os autores também mencionam que, baseado na informação de previsão, operadores podem empreender ações preventivas ou corretivas em operações seguras e econômicas em sistemas elétricos de potência. Dessa forma, os autores criaram um modelo baseado em regressão linear para ser usado em sistemas de previsão de demanda de energia elétrica. Eles relatam que esse modelo foi aplicado em uma empresa americana do ramo de geração e distribuição de energia elétrica e comprovam, através da estimação dos erros de previsão, que seu desempenho foi bem superior ao modelo anteriormente empregado.

A previsão precisa da energia fotovoltaica pode ser um processo complexo tarefa devido à natureza flutuante do clima (por exemplo, nuvem em movimento e as mudanças de temperatura). Recentemente, redes neurais recorrentes foram usadas em várias aplicações alcançando bons resultados em diferentes aplicações, porque podem modelar a dinâmica dos dados. Em Abdel-Nasser e Mahmoud (2017) propõe-se o uso de redes neurais de memória de longo prazo (*Long Short-Term Memory Recurrent Neural Network - LSTM RNN*) para previsão da produção fotovoltaica de energia. O LSTM-RNN pode modelar as mudanças temporais nos dados devido à sua arquitetura recorrente e unidades de memória. Ao contrário das redes neurais recorrentes tradicionais, os LSTM foram projetados para evitar o problema da dependência a longo prazo. De fato, o LSTM pode capturar conceitos abstratos nas sequências de geração de energia fotovoltaicas. Neste trabalho, o uso de LSTM-RNN também é considerado visando obter uma adequada previsão da produção de energia de uma usina fotovoltaica.

Para Su, Batzelis e Pal (2019) há uma grande gama de algoritmos de aprendizado de máquina (*Machine Learning*) na previsão de geração de energia fotovoltaica. Uma revisão da literatura mostra que as abordagens de aprendizado de máquina são geralmente superiores aos métodos estatísticos convencionais devido à sua habilidade inerente para modelar não-linearidades, processos complexos e dinâmicos. Contudo, o treinamento de uma RNA (Rede Neural Artificial) ou de uma IA (Inteligência Artificial) é complicado e ainda não existe uma maneira comumente aceitável de construir um modelo perfeito; por isso é que selecionar e otimizar os parâmetros do modelo geralmente é um processo de tentativa e erro. A maioria dos estudos relevantes na literatura examina apenas algumas metodologias de aprendizado de máquina, com foco em previsões de curto prazo (até três dias antes) e não fornecendo detalhes suficientes sobre como os parâmetros do modelo são encontrados; uma análise comparativa abrangente para dar conta de todos os métodos relevante e tempos de previsão mais longos ainda está faltando na literatura. Além disso, os vários estudos consideram diferentes instalações do mundo real com especificações diferentes da planta, locais, períodos, condições climáticas e conjuntos de dados, enquanto não há maneira consistente de selecionar as variáveis de treinamento do modelo e métricas

de erro. Até hoje, esses métodos ainda não foram avaliados em uma estrutura de avaliação comum simultaneamente.

Su, Batzelis e Pal (2019) implementou uma comparação entre dez algoritmos diferentes de aprendizado de máquina para a previsão de energia fotovoltaica com seis dias de antecedência. Esses algoritmos incluem seis Redes Neurais Recorrentes (*Recurrent Neural Network*) e quatro métodos de AI, como segue:

- *Back Propagation Neural Network - BPNN*;
- *BPNN with Genetic Algorithm Optimization*;
- *Elman Neural Network - ENN*;
- *Generalized Regression Neural Network - GRNN*;
- *Adaptive Network based Fuzzy Inference System - ANFIS*;
- *Nonlinear Autoregressive Neural Network with Exogenous Inputs - NARXNN*;
- *K Nearest Neighbours - KNN*;
- *Extreme Learning Machine - ELM*;
- *Random Forest - RF*;
- *Support Vector Regression - SVM*.

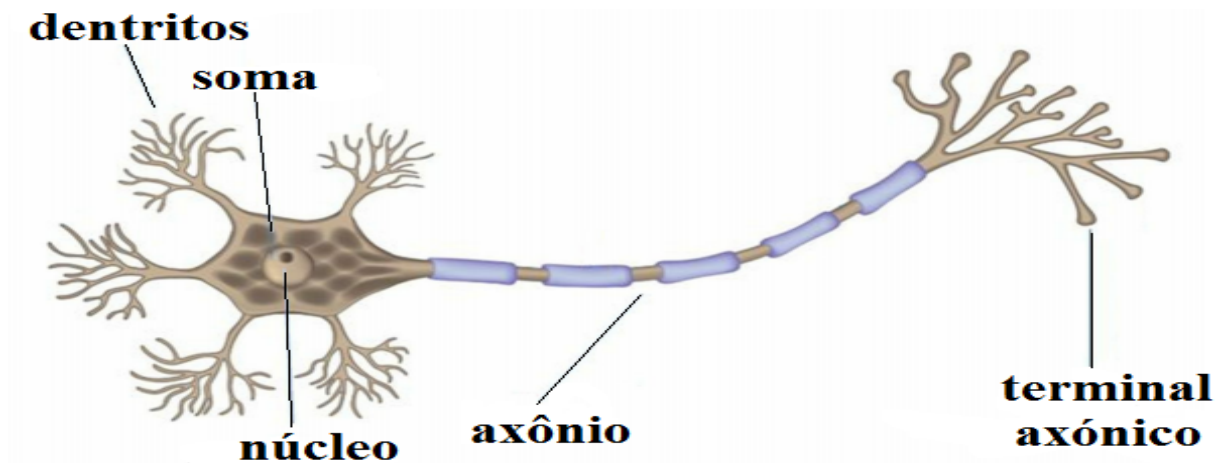
## 2.4 Redes neurais

A primeira publicação relacionada à neurocomputação data de 1943, por meio do artigo elaborado por McCulloch e Pitts. Neste trabalho de McCulloch e Pitts (1943), os autores realizaram o primeiro modelamento matemático inspirado no neurônio biológico, resultando assim, na primeira concepção de neurônio artificial. Segundo Silva (2016), o primeiro método de treinamento para redes neurais artificiais foi proposto Hebb (1949), e se denominou de regra de aprendizado de Hebb, sendo esta baseado em hipóteses e observações de caráter neurofisiológico.

### 2.4.1 Neurônio biológico

A célula elementar do sistema nervoso cerebral é o neurônio e seu papel se resume a conduzir impulsos (estímulos elétricos advindos de reações físico-químicas) sob determinadas condições de operação. Segundo Silva (2016) tal elemento biológico pode ser dividido em três partes principais, isto é, nos dendritos, no corpo celular (também conhecido como soma) e o axônio. A Figura 2.3 apresenta a estrutura de um neurônio biológico.

Figura 2.3 – Estrutura de um neurônio.



Fonte: Adaptado(CIABURRO; VENKATESWARAN, 2017).

Segundo Ciaburro e Venkateswaran (2017), cada um dos elementos citados dendritos, corpo celular (soma) e axônios, têm função específica:

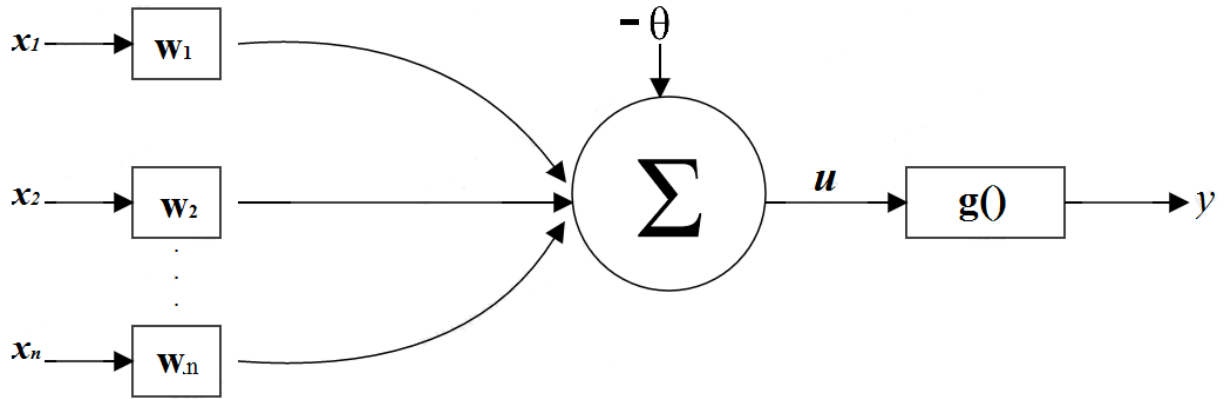
- Dendritos: pontos de entrada em cada neurônio que recebem informações de outros neurônios da rede em forma de impulsos elétricos;
- Corpo celular (Soma): gera inferências a partir das entradas de dendritos e decide qual ação deve-se realizar;
- Axônios: eles transmitem saídas em forma de impulsos elétricos para o próximo neurônio.

#### 2.4.2 Neurônio artificial

Segundo Silva (2016) a estrutura das redes neurais artificiais foi desenvolvida a partir de modelos conhecidos de sistemas nervosos biológicos e do próprio cérebro humano. Os elementos computacionais ou unidades processadoras, denominadas neurônios artificiais, são modelos bem simplificados dos neurônios biológicos. Ainda em Silva (2016) os neurônios artificiais utilizados nos modelos de redes neurais artificiais são não-lineares, fornecem saídas tipicamente contínuas, e realizam funções simples, como coletar os sinais existentes em suas entradas, agregá-los de acordo com sua função operacional e produzir uma resposta, levando em consideração sua função de ativação inerente.

O modelo de neurônio mais simples e que engloba as principais características de uma rede neural biológica, isto é, paralelismo e alta conectividade de acordo com McCulloch e Pitts (1943), é mostrado na Figura 2.4. Segundo Silva (2016), esse é ainda o modelo mais utilizado nas diferentes arquiteturas de redes neurais artificiais.

Figura 2.4 – Modelo de neurônio artificial.



Fonte: Adaptado(SILVA, 2016).

Para Silva (2016), depreende-se da Figura 2.4, que o neurônio artificial funciona a partir de elementos básicos como:

- Sinais de entrada  $\{x_1, x_2, \dots, x_n\}$ : sinais advindos do meio externo e que representam os valores assumidos pelas variáveis de uma aplicação específica;
- Pesos sinápticos  $\{W_1, W_2, \dots, W_n\}$ : valores que servirão para ponderar cada uma das variáveis de entrada da rede, permitindo-se quantificar as suas relevâncias em relação à funcionalidade do respectivo neurônio;
- Combinador linear  $\Sigma$ : sua função é agregar todos os sinais de entrada que foram ponderados pelos respectivos pesos sinápticos a fim de produzir um valor de potencial de ativação;
- Limiar de ativação  $\theta$ : variável que especifica qual será o patamar apropriado para que o resultado produzido pelo combinador linear possa gerar um valor de disparo em direção à saída do neurônio;
- Potencial de ativação  $u$ : obtido pela diferença do valor produzido entre o combinador linear e limiar de ativação. Se tal valor é positivo, ou seja, se  $u \geq \theta$  então o neurônio produz um potencial excitatório; caso contrário, o potencial será inibitório;
- Função de ativação  $g$ : seu objetivo corresponde a limitar a saída do neurônio dentro de um intervalo de valores razoáveis a serem assumidos pela sua própria imagem funcional;
- Sinal de saída  $y$ : valor final produzido pelo neurônio em relação a um determinado conjunto de sinais de entrada, podendo ser também utilizado por outros neurônios sequencialmente interligados.

Os diversos sinais de entrada advindos do meio externo, são rerepresentados pelo conjunto de variáveis  $\{x_1, x_2, \dots, x_n\}$ , que são análogos aos impulsos elétricos externos captados pelos dendritos no neurônio biológico.

Em Silva (2016), destaca ainda que ponderações exercidas pelas junções sinápticas do modelo biológico são representadas no neurônio artificial pelo conjunto de pesos sinápticos  $\{W_1, W_2, \dots, W_n\}$ . De forma análoga, a relevância de cada uma das entradas  $x_i$  do neurônio é então executada por meio de suas multiplicações pelos respectivos pesos sinápticos  $W_i$ ,  $1 \leq i \leq n$ , ponderando-se, portanto, todas as informações externas que chegam ao neurônio. Assim, torna-se possível verificar que a saída do corpo celular artificial, denotado por  $u$ , é a soma ponderada das entradas.

As expressões seguintes sintetizam o resultado produzido pelo modelo de neurônio artificial de McCulloch e Pitts (1943).

$$u = \sum_{i=1}^n W_i \cdot x_i - \theta \quad (2.1)$$

$$y = g(u) \quad (2.2)$$

### 2.4.3 Entendendo como uma rede neural artificial funciona

Pode-se dizer que uma rede neural artificial consiste de um grande número de elementos de processamento simples chamados neurônios, unidades, células ou nós. Cada neurônio é conectado a outro neurônio por meio de links diretos de comunicação, cada um deles com um peso associado. Os pesos representam a informação sendo usada pela rede para resolver um problema. Redes neurais podem ser aplicadas a uma grande variedade de problemas, tais como armazenamento e recuperação de dados ou padrões, classificação de padrões, realizar mapeamento geral dos padrões de entrada para os padrões de saída, agrupando padrões similares, ou encontrar soluções para problemas de otimização com restrições.

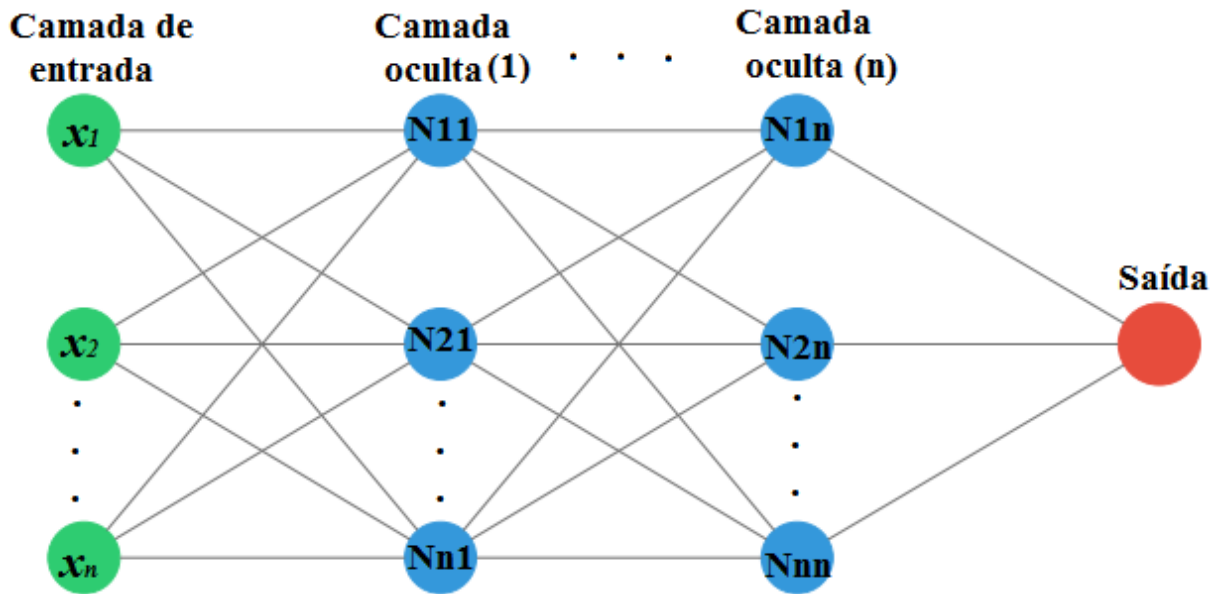
Segundo Fausett (1993), uma rede neural caracteriza-se por:

- padrão de conexões entre os neurônios (chamado de *arquitetura*);
- método para determinar os pesos nas conexões (chamado de seu *treinamento*, ou *aprendizado*, *algoritmo*);
- *função de ativação*.

### 2.4.4 Arquitetura

Para Silva (2016), a arquitetura é a forma como as unidades fundamentais, ou seja, os neurônios, estão arranjados uns em relação aos outros. Silva (2016) divide uma rede neural artificial, como da Figura 2.5, em três partes:

Figura 2.5 – Rede neural artificial.



Fonte: O autor.

- Camada de entrada: responsável pelo recebimento de dados, sinais ou medições advindas do meio externo, sendo que as entradas são geralmente normalizadas em relação às faixas de variação dinâmicas produzidas pelas funções de ativação. Esta normalização implica numa melhor precisão numérica frente às operações matemáticas realizadas pela rede;
- Camadas ocultas: são compostas de neurônios que possuem a responsabilidade de extrair as características associadas ao processo ou sistema a ser inferido. Quase todo o processo interno da rede é realizado nessas camadas;
- Camada de saída: constituída de neurônios, sendo responsável pela produção e apresentação dos resultados finais da rede, os quais são advindos dos processos efetuados pelos neurônios das camadas anteriores.

Observe que  $x_1, x_2, \dots, x_n$  são as entradas e  $N_{11}, N_{21}, \dots, N_{n1}$  são os neurônios da camada (1), assim como  $N_{1n}, N_{2n}, \dots, N_{nn}$  são os neurônios da camada (n).

Ainda para Silva (2016) as principais arquiteturas são:

- Arquitetura *feedforward* de camada simples;
- Arquitetura *feedforward* de camadas múltiplas;
- Arquitetura recorrente ou realimentada;
- Arquitetura em estrutura reticulada.

No presente trabalho foi utilizada a arquitetura recorrente, que são redes em que as saídas dos neurônios são realimentadas como sinais de entrada para outros neurônios.

### 2.4.5 Processo de treinamento

Para Silva (2016) o processo de treinamento de uma rede neural consiste na aplicação de passos ordenados que sejam necessários para a sintonização dos pesos sinápticos e limiares de seus neurônios, tendo-se como objetivo final a generalização de soluções a serem produzidas pelas suas saídas, cujas respostas são representativas do sistema físico em que estas estão sendo mapeando. O conjunto desses passos ordenados visando o treinamento da rede é denominado de algoritmo de aprendizagem. Ao longo de suas aplicação, a rede será então capaz de extrair características discriminantes do sistema a ser mapeado por intermédio de amostras que foram retiradas de seu contexto. Ainda segundo Silva (2016), as estratégias de treinamento podem ser:

- Treinamento supervisionado;
- Treinamento não-supervisionado;
- Treinamento com reforço;
- Aprendizagem usando lote de padrões (*off-line*);
- Aprendizagem usando padrão-por-padrão (*on-line*).

### 2.4.6 Função de ativação

Como já foi mencionado, a função de ativação tem como objetivo limitar a saída do neurônio. Pode-se dizer que há dois tipos de funções de ativação:

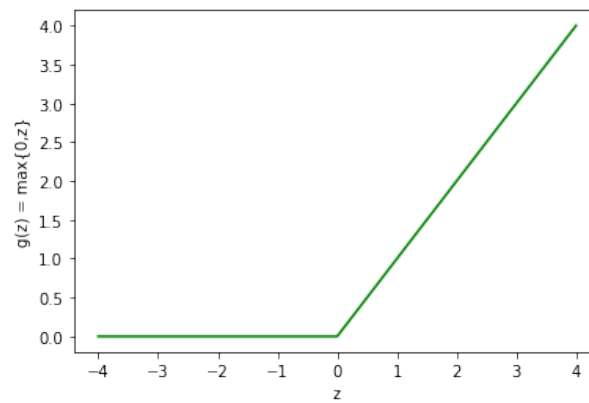
- Funções de ativação parcialmente diferenciáveis: aquelas que possuem pontos cujas derivadas de primeira ordem são inexistentes, são exemplos: função degrau, função sinal, função rampa simétrica;
- Funções totalmente diferenciáveis: são aquelas cujas derivadas de primeira ordem existem e são conhecidas em todos os pontos de seu domínio de definição. São



exemplos: função logística, função tangente hiperbólica, função gaussiana, função linear.

Segundo Goodfellow, Bengio e Courville (2017), nas redes neurais modernas, a recomendação padrão é usar a função *rectified linear unit* - *ReLU*. Neste trabalho foram usadas duas funções de ativação diferentes para a camada de entrada, as camadas ocultas e a camada de saída. A função de ativação da camada de entrada e das camadas ocultas foi a *ReLU*, mostrada na Figura 2.6.

Figura 2.6 – Função ReLU.

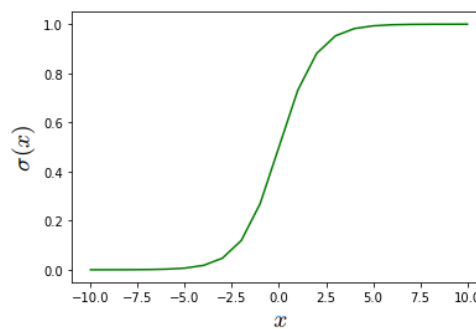


Fonte: O autor.

Como função de ativação para a camada de saída foi usada a função logística sigmoide (*logistic sigmoid*), mostrada na Figura 2.7.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \quad (2.3)$$

Figura 2.7 – Função sigmoide logística.

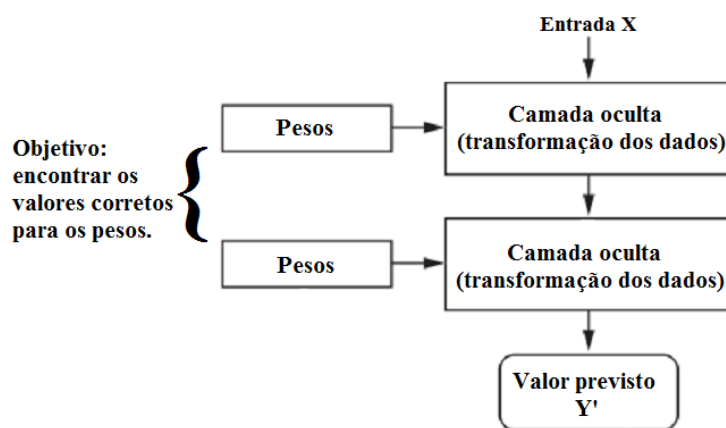


Fonte: O autor.

### 2.4.7 Ciclo de treinamento

A especificação do que uma camada faz com seus dados de entrada é armazenada nos pesos, que em essência são um monte de números. Em termos técnicos, diríamos que a transformação implementada por uma camada é parametrizada por seus pesos (2.8), sendo que às vezes os pesos também são chamados de parâmetros de uma camada. Segundo Ciaburro e Venkateswaran (2017) nesse contexto, o aprendizado significa encontrar um conjunto de valores para os pesos de todas as camadas em uma rede, de modo que a rede mapeará corretamente exemplos de entradas para seus destinos associados. Mas aqui está o problema: uma rede neural profunda pode conter dezenas de milhões de parâmetros. Encontrar o valor correto para todos eles pode parecer uma tarefa assustadora, especialmente considerando que modificar o valor de um parâmetro afetará o comportamento de todos os outros.

Figura 2.8 – Uma rede neural é parametrizada pelos seus pesos.

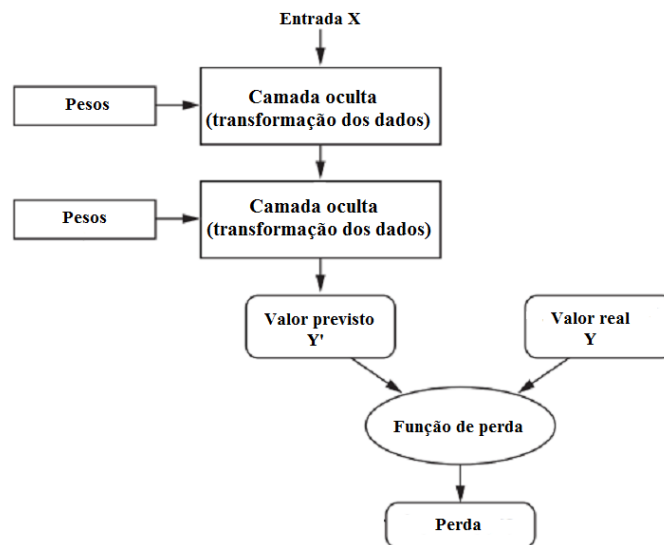


Fonte: Adaptado(CHOLLET, 2018).

Para Chollet (2018), para controlar a saída de uma rede neural, é preciso medir o erro dessa saída em relação ao valor esperado. Esse é o trabalho da função de perda da rede. A função de perda toma as previsões da rede e o verdadeiro alvo (o que se deseja que a rede produza) e calcula o erro de distância, capturando quão bem a rede se saiu, um exemplo desse processo é mostrado na Figura 2.9. A Figura 2.10 mostra uma rede neural com características um pouco diferentes à mostrada na Figura 2.9.

Inicialmente, os pesos da rede recebem valores aleatórios, assim a rede apenas implementa uma série de transformações aleatórias. Naturalmente, sua saída está longe do que deveria ser idealmente, e a perda é, portanto, muito alta. Mas com cada exemplo que a rede processa, os pesos são ajustados um pouco na direção correta e a perda diminui, isto é, usa-se um sinal de *feedback* para ajustar um pouco o valor dos pesos, em uma direção que diminua a perda. Esse ajuste é o trabalho do otimizador, que implementa o que é chamado algoritmo de Retro-propagação (*Backpropagation*). A Figura 2.10 mostra uma

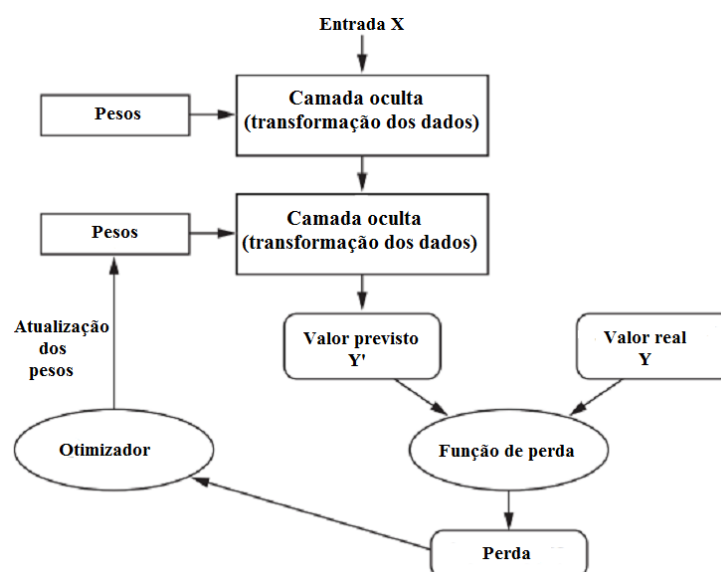
Figura 2.9 – A função de perda mede a qualidade da saída da rede.



Fonte: Adaptado(CHOLLET, 2018).

rede neural com as características citadas. Esse é o ciclo de treinamento, é repetido um número suficiente de vezes (geralmente dezenas de iterações em milhares de exemplos), cada repetição usando todo conjunto de amostras para treinamento é chamada de época de treinamento, ou apenas época, isso gera valores de peso que minimizam a função de perda. Uma rede treinada é então, uma rede com uma perda mínima e para qual as saídas estão os mais próximos possível dos valores alvos.

Figura 2.10 – A perda é usada como sinal de realimentação para ajustar os pesos.



Fonte: Adaptado(CHOLLET, 2018).

## 2.5 Memória de longo prazo

O tipo de rede que será utilizado neste trabalho para previsão de geração fotovoltaica corresponde a uma rede com memória de longo prazo (ou *Long Short-Term Memory* - LSTM). A LSTM é um tipo de Rede Neural Recorrente (*Recurrent Neural Network* - RNN), segundo Brownlee (2017), RNN são um tipo especial de rede neural projetadas para problemas de sequência. Considerando uma rede *perceptron* multicamada (*Multi Layer Perceptron* - MLP), uma RNN pode ser pensada como a adição de *loops* de arquitetura. Por exemplo, em uma dada camada, cada neurônio pode passar seu sinal lateralmente além de passar para a próxima camada. A saída da rede pode realimentar à entrada a rede com o próximo vetor de entrada. E assim por diante.

Segundo Brownlee (2017), a rede LSTM é diferente da clássica MLP. Em uma MLP, a rede é composta de camadas de neurônios. Entrada de dados é propagada através da rede para fazer a previsão.

Como parte das RNN, uma LSTM tem conexões recorrentes de tal forma que o estado de ativação do neurônio do passo anterior é usado como contexto para formulação de uma saída. Mas diferente de outras RNN, a LSTM tem uma formulação única que permite evitar os problemas que impedem o treinamento de outros tipos de RNN. Isto, e os impressionantes resultados que podem ser alcançados, são a razão da popularidade da técnica.

### 2.5.1 Célula de memória LSTM

Segundo Brownlee (2017), a unidade computacional de uma rede LSTM é chamada célula de memória (*memory cell*), bloco de memória (*memory block*), ou apenas célula (*cell*). O termo neurônio como unidade computacional está arraigado ao uso de MLPs que é também frequentemente usado para se referir as células de memória de uma LSTM. As células LSTM são compostas de pesos e portões(*gates*).

Na fase de aprendizagem, redes neurais tradicionais não utilizam a informação aprendida na etapa anterior no modelamento dos dados no passo atual. Este ponto representa uma grande falha das redes neurais tradicionais. RNNs tentam resolver esse problema usando *loops* que passam informações de uma etapa da rede para as próximas etapas, permitindo que a informação persista. Em outras palavras, RNNs conectam as informações anteriores à tarefa atual. De fato, o uso de amostras de sequência anteriores pode ajudar no entendimento da presente amostra (ABDEL-NASSER; MAHMOUD, 2017).

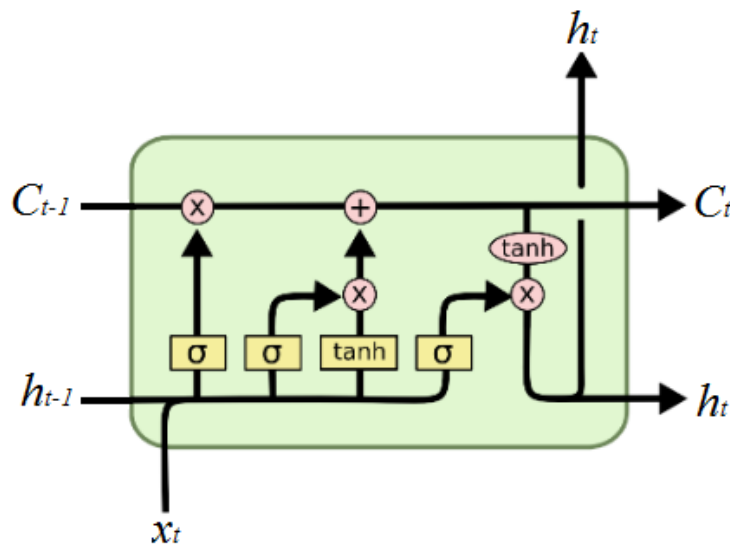
Ao contrário dos RNN, as LSTM foram projetados para evitar o problema da dependência a longo prazo. A rede LSTM é treinada usando retro propagação através do tempo, e supera o problema de dissipação do gradiente (*vanishin gradient*). O problema de dissipação do gradiente é uma dificuldade encontrada no treinamento de RNA, esse

problema torna muito difícil aprender e ajustar os parâmetros das camadas anteriores da rede. Esse problema se agrava à medida que o número de camadas na arquitetura aumenta. As redes neurais tradicionais têm neurônios, por outro lado, as redes LSTM possuem células de memória conectados por camadas sucessivas. Cada célula contém portões que lidam com o estado da célula e da saída. Segundo Abdel-Nasser e Mahmoud (2017), no célula LSTM, há três tipos de *gates*: *forget*, *input*, e *output*. A tarefa de cada *gate* pode ser resumida como segue:

- *Gate* de esquecimento (*Forget gate*): define quais informações podem ser excluídas da célula com base em determinadas condições;
- *Gate* de entrada (*Input gate*): define quais valores da entrada devem ser atualizados para a memória de estado com base em determinadas condições;
- *Gate* de saída (*Output gate*): define o que enviar com base na entrada e na memória da célula com base em determinadas condições.

Na Figura 2.11, há um diagrama de um célula de memória LSTM, o funcionamento desta célula será apresenta no decorrer deste trabalho.

Figura 2.11 – Célula de memória LSTM.



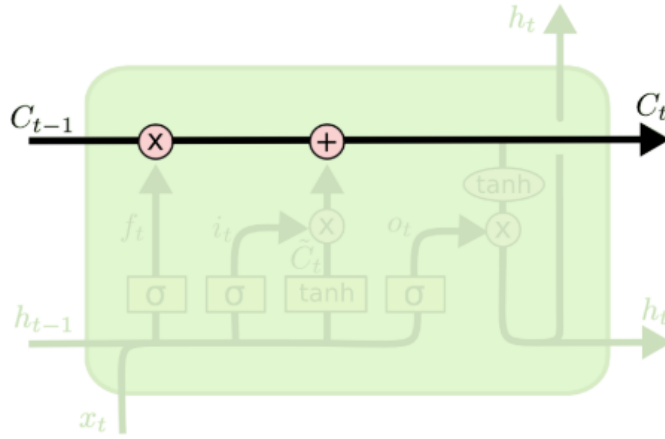
Fonte: Adaptado(OLAH, 2015)

## 2.5.2 Entendendo a célula de memória LSTM

Segundo Olah (2015), a chave para compreender a célula de memória LSTM é a linha horizontal percorrendo a parte superior do diagrama da Figura 2.12, o estado da célula (ou bloco) de memória LSTM. O estado da célula é como uma esteira transportadora.

Ele percorre toda a cadeia, com apenas algumas interações lineares menores. É muito fácil que as informações fluam inalteradas. Os *gates* regulam a forma como as redes LSTM

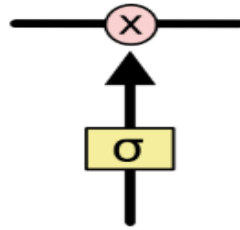
Figura 2.12 – Estado da célula de memória LSTM.



Fonte: (OLAH, 2015).

removem ou adicionam informação ao estado da célula. Segundo Olah (2015) os *gates* são uma maneira otimizada de deixar a informação passar. Os *gates* de forma genérica são compostos de uma camada de rede neural sigmoide e uma operação de multiplicação, como na Figura 2.13. A camada sigmoide gera números entre zero e um, descrevendo quanto de

Figura 2.13 – *Gate* genérico.

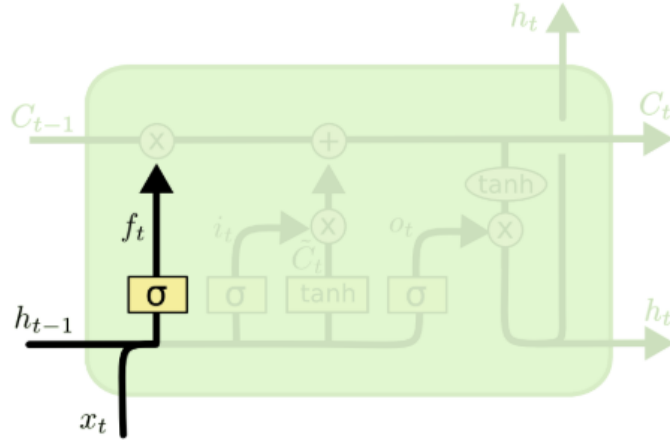


Fonte: (OLAH, 2015).

cada componente, da informação, deve ser liberado. Um valor zero significa que nada irá passar, enquanto um valor de um significa que tudo irá passar.

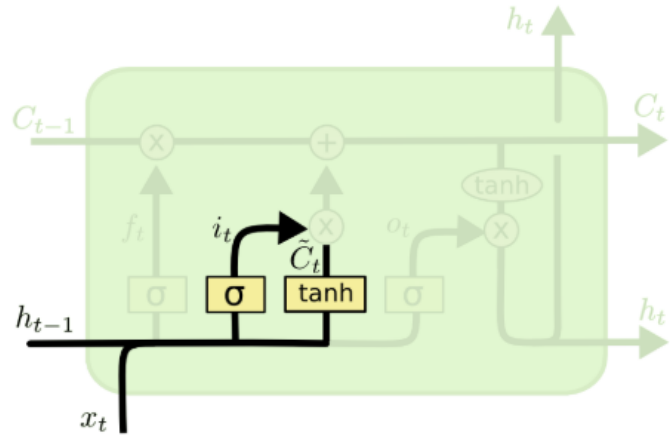
Para Olah (2015), inicialmente a rede LSTM deve decidir quais informações serão jogadas fora do estado da célula. Essa decisão é tomada pelo *forget gate*. O *forget gate* recebe como entradas  $h_{t-1}$  e  $x_t$ , em forma de vetor  $[h_{t-1}, x_t]$ , produzindo uma saída entre 0 e 1 para cada valor  $C_{t-1}$ , estado da célula de memória anterior. Assim, o valor 1 significa que o valor atual do estado da célula será mantido, já valor 0 significa que o valor atual será descartado. A Figura 2.14 e a equação 2.4 mostram a atuação do *forget gate*.

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (2.4)$$

Figura 2.14 – Atuação do *forget gate*.

Fonte: Adaptado(OLAH, 2015).

Ainda segundo Olah (2015) o próximo passo é decidir qual informação será guardada no estado da memória. Primeiro o *input gate* decide quais valores serão atualizados, essa decisão ocorre como no caso do *forget gate*, o *input gate* recebe o vetor  $[h_{t-1}, x_t]$  e fornece uma saída entre 0 e 1. A saída do *input gate* é multiplicada pelo valor do vetor  $\tilde{C}_t$  (ou  $\bar{C}_t$ ), esse vetor será usado para atualizar o estado da célula. A Figura 2.15 e as equações 2.5 e 2.6 exemplificam o processo anterior.

Figura 2.15 – Atuação do *input gate*.

Fonte: Adaptado(OLAH, 2015).

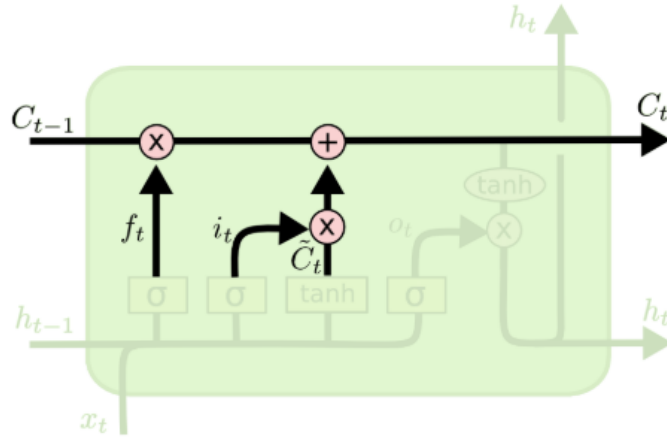
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (2.5)$$

$$\bar{C} = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \quad (2.6)$$

Agora deve-se atualizar o antigo valor de estado da célula,  $C_{t-1}$ , para um novo valor  $C_t$ . Para isso, multiplica-se o valor antigo do estado da célula por  $f_t$ , esquecendo

assim o valor anterior. Em seguida, adiciona  $i_t \cdot \bar{C}_t$ . Esse processo está ilustrado na Figura 2.16 e na equação 2.7.

Figura 2.16 – Atualização do estado da célula.

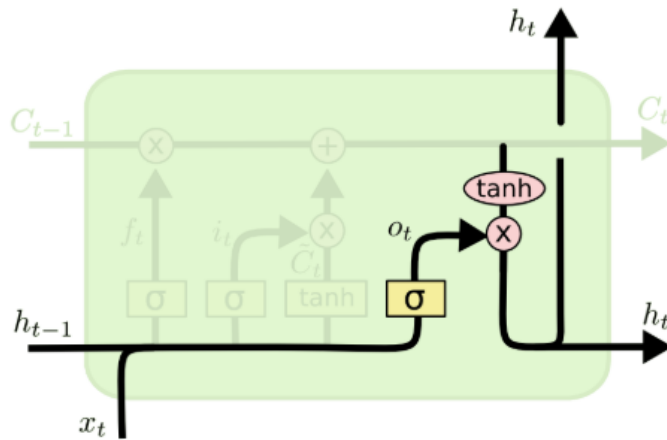


Fonte: Adaptado(OLAH, 2015).

$$C_t = f_i \cdot C_{t-1} + i_t \cdot \bar{C}_t \quad (2.7)$$

Por fim segundo Olah (2015), é necessário decidir o que será a saída. A saída é baseada no estado da célula, mas será um valor filtrado, isso porque ainda passar pelo *output gate*. Primeiro, assim como no *forget gate*, o *output gate* recebe o vetor  $[h_{t-1}, x_t]$  e fornece uma saída entre 0 e 1. Em seguida a *tanh* recebe o  $C_t$  e então o valor do *output gate* é multiplicado pelo valor de  $\tanh(C_t)$ . Esse processo é ilustrado pela Figura 2.17 e equações 2.8 e 2.9.

Figura 2.17 – Atuação do *output gate*.



Fonte: Adaptado(OLAH, 2015).

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \quad (2.8)$$



$$h_t = o_t \cdot \tanh(C_t) \quad (2.9)$$

O pseudo algoritmo a seguir exemplifica o funcionamento de uma célula de memória LSTM como uma função:

1	<b>função</b> célula-LSTM( $C_{t-1}, h_{t-1}, x_t$ ):
2	$combinar = [h_{t-1}, x_t]$
3	$f_t = \sigma(W_f \cdot combinar + b_f)$
4	$i_t = \sigma(W_i \cdot combinar + b_i)$
5	$\bar{C} = \tanh(W_C \cdot combinar + b_C)$
6	$C_t = f_t \cdot C_{t-1} + i_t \cdot \bar{C}_t$
7	$o_t = \sigma(W_o \cdot combinar + b_o)$
8	$i_t = \sigma_t \cdot \tanh(C_t)$
9	$h_t = o_t \cdot \tanh(C_t)$
10	retornar $h_t, C_t$



## 3 Metodologia para previsão de produção de usina PV

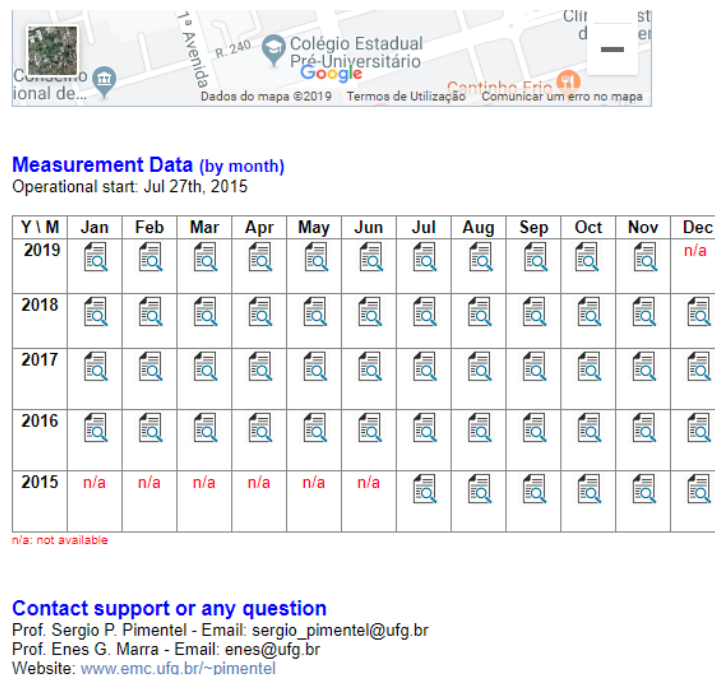
### 3.1 Manipulação dos dados

Para este trabalho foram utilizados dois conjuntos de dados, dados de entrada e dados de saída, obtidos de duas fontes distintas. Os dados de entrada foram obtidos da estação meteorológica situada no bloco B da Escola de Engenharia Elétrica, Mecânica e de Computação (EMC) da Universidade Federal de Goiás (UFG). Os dados de saída foram obtidos do sistema fotovoltaico conectado à rede nas dependências da EMC.

#### 3.1.1 Coleta

Os dados de entrada foram obtidos do site da estação meteorológica da EMC, através de uma interface como a mostrada na Figura 3.1.

Figura 3.1 – Site da estação meteorológica localizada na EMC.



Fonte: Adaptado(MARRA; PIMENTEL, 2015).

Foi feito download das planilhas, mostrada na Figura 3.2, com os dados meteorológicos de março a setembro 2019. Depois de uma análise visual constatou-se que dados como *Wind Speed*, *Wind Direction*, *Precipitation* tinham várias inconsistências, principalmente dados simultaneamente nulos e ausência de dados. Assim, optou-se de forma inicial por

Figura 3.2 – Planilha de dados meteorológicos março de 2019.

2019\_03\_EMC-UFG\_Drive

Ficheiro Editar Ver Inserir Formatar Dados Ferramentas Suplementos Ajuda

100% Apenas visualização

	A	B	C	D	E	F	G	H	I
1	Instante da medida	Registro	Velocidade do vento	Direção do vento	Temp. Ambiente	Umid. Relat. do Ar	Índice Pluviométrico	Irradiância Solar	Press. Atmosférica
2	Measurement Time	RECORD	Wind Speed	Wind Direction	Air Temperature	Relative Humidity	Precipitation	Solar Irradiance	Atm. Pressure
3	mm/dd/yy hh:mm	#	m/s	° (0=North, CW)	°C	%	mm	W/m <sup>2</sup> – Global	mBar
4	3/1/19 0:00	1707213	0.7416667	298.89	22.02	89.73	0	2.743735	939.11
5	3/1/19 0:01	1707214	0.6083333	0.68	21.99	89.85	0	2.866923	939.43
6	3/1/19 0:02	1707215	1.0625	285.66	21.98	89.82	0	2.932516	939.4
7	3/1/19 0:03	1707216	0.825	328.64	21.96	89.85	0	2.814128	939.17
8	3/1/19 0:04	1707217	1.2	347.8	21.93	89.92	0	2.866923	939.16

Fonte: Adaptado(MARRA; PIMENTEL, 2019).

não considerar tais dados na precisão proposta no presente trabalho. Sendo assim, as variáveis, não nulas, inicialmente serão consideradas na entrada foram : *Air Temperature* , *Relative Humidity*, *Solar Irradiance*, *Atm. Pressure*. Observou-se também que as medições dessas variáveis ocorria a cada minuto.

Já os dados de saída foram obtidos da saída do sistema fotovoltaico nas dependência da EMC. Para tanto, pesquisou-se no site do fabricante dos inversores (o fabricante dos inversores fornece um endereço IP para monitoramento dos dados), ver Figura 3.3, informações sobre a geração do sistema como um todo e de seus inversores individuais. Contudo os dados tiveram que ser transcrito manualmente, ponto a ponto do gráfico diário, em que cada ponto representa geração em um intervalo de 15 minutos.

Figura 3.3 – Site do sistema de monitoramento conectado a rede.



Fonte: Adaptado(ANALYZER, 2019).

Optou-se por coletar dados no intervalo das 06:00 as 18:00 horas. A justificativa

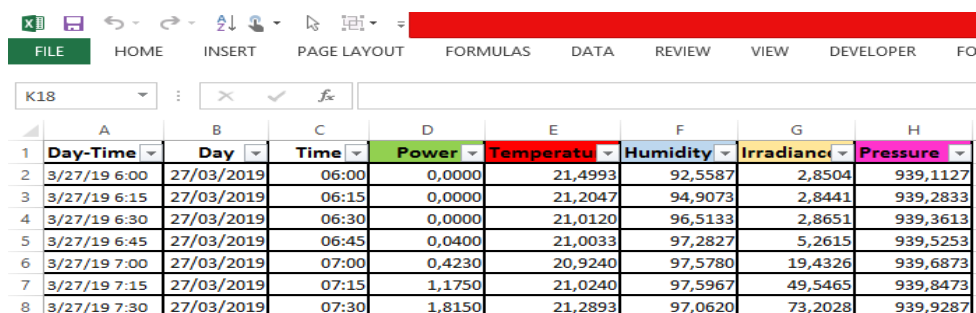
para isso dá-se pela representatividade dos dados coletados inicialmente, que apresentavam geração apenas entre esse intervalo do dia. Os dados foram coletados manualmente e listados em uma planilha.

### 3.1.2 Depuração dos dados

O depuramento dos dados de entrada, como citado, inicialmente foi visual, eliminando valores das variáveis que apresentavam inconsistências. Em um passo posterior devido aos dados de saída serem coletados a cada 15 minutos e os dados de entrada a cada minuto, fez-se a média dos dados de entrada em intervalos de 15 minutos, 60 minutos e  $60 \times 12 = 720$  minutos, em um período de tempo das 6 as 18 horas. Para fazer a média foi implementada uma macro em *Visual Basic* do editor de planilhas *Excel*. Nesse ponto, notou-se que em alguns meses haviam inconsistências no intervalo de tempo amostrado. Observou-se que para a quantidade de dias do mês, em média de 30 dias, tomado o intervalo de 6 as 18 horas e realizando 4 medições por hora, um total de  $4 \times 12 \times 30 = 1440$  medições mensais, em alguns meses o total de medições era menor que em um mês médio ou maior que em um mês de 31 dias. Descobriu-se que havia dias, horas, minutos faltando em alguns meses, o que diminuía o número medições. As vezes minutos estavam replicados, o que aumentava o número de medições.

Os dias em que haviam horas ou minutos faltando foram eliminados da lista de entrada, exceto aqueles em que havia apenas um minuto faltando dentro do intervalo das 6 as 18 horas, nestes fez-se a média dos 3 minutos antes e depois do minuto que faltava. Aqueles em que o minuto estava replicado apenas eliminou-se o elemento repetido. Em seguida utilizando as ferramentas de filtragem do editor filtraram-se os dados em intervalos de tempo de 15 minutos (*data1*), 60 minutos (*data2*) e 720 minutos (*data3*), dentro do intervalo de 6 as 18 horas, obtendo-se assim três conjuntos iniciais de dados de entrada com as seguintes variáveis de entrada *Temperature*, *Humidity*, *Irradiance*, *Pressure* e um conjunto de saída chamado de *Power*. Parte do conjunto *data1* está representado na tabela da Figura 3.4.

Figura 3.4 – Dados de entrada para médias de 15 minutos.

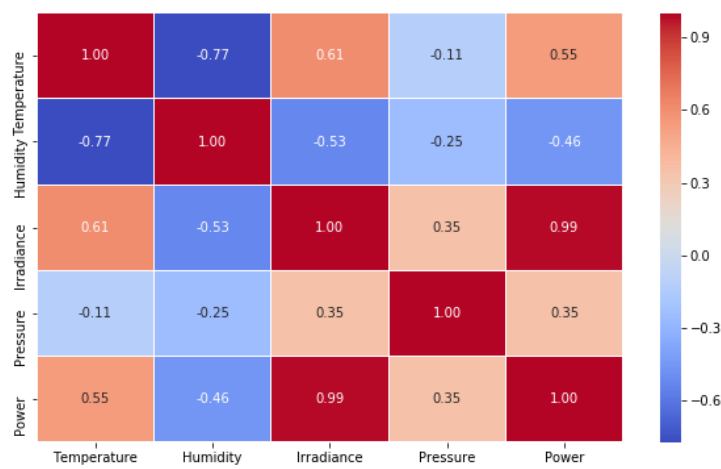


	A	B	C	D	E	F	G	H
	Day-Time	Day	Time	Power	Temperatu	Humidity	Irradianc	Pressure
2	3/27/19 6:00	27/03/2019	06:00	0,0000	21,4993	92,5587	2,8504	939,1127
3	3/27/19 6:15	27/03/2019	06:15	0,0000	21,2047	94,9073	2,8441	939,2833
4	3/27/19 6:30	27/03/2019	06:30	0,0000	21,0120	96,5133	2,8651	939,3613
5	3/27/19 6:45	27/03/2019	06:45	0,0400	21,0033	97,2827	5,2615	939,5253
6	3/27/19 7:00	27/03/2019	07:00	0,4230	20,9240	97,5780	19,4326	939,6873
7	3/27/19 7:15	27/03/2019	07:15	1,1750	21,0240	97,5967	49,5465	939,8473
8	3/27/19 7:30	27/03/2019	07:30	1,8150	21,2893	97,0620	73,2028	939,9287

Fonte: O autor.

Em uma etapa posterior, foi realizado um teste de correlação entre todas as variáveis de entrada com a variável de saída, mostrado na Figura 3.5. Pode-se observar que há uma correlação positiva muito forte, 0.99, entre a variável *Irradiance* e a variável de saída *Power* para o conjunto de dados de médias em hora (*data2*). Essa correlação era esperada, já que as variáveis possuem uma relação causal entre si. Contudo, ela é maior que a correlação dos outros conjuntos (*data1*, *data3*). Todas as outras variáveis não apresentaram correlação significativa. No entanto como veremos adiante elas parecem influenciar o treinamento do modelo.

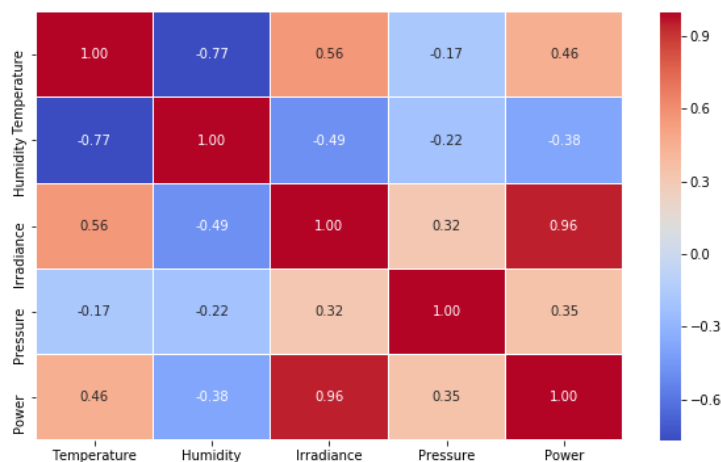
Figura 3.5 – Teste de correlação *data2*.



Fonte: O autor.

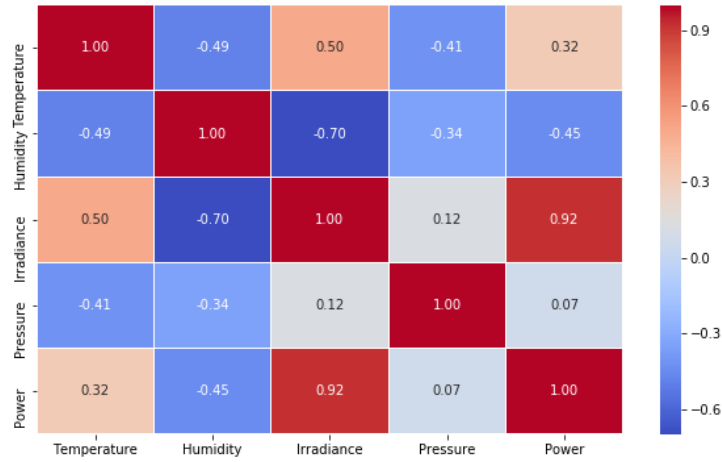
Assim como exposto anteriormente, observou-se que as maiores correlações obtidas foram 0,96 positiva (*data1*), mostrado na Figura 3.6, e 0,92 positiva (*data3*), mostrado na Figura 3.7. Não pretende-se fazer aqui uma relação causal entre uma correlação forte entre

Figura 3.6 – Teste de correlação, conjunto *data1*.



Fonte: O autor.

Figura 3.7 – Teste de correlação, conjunto data3.



Fonte: O autor.

as variáveis e as previsões do modelo adotado, contudo como será visto posteriormente o modelo teve melhores resultados para o conjunto *data2*.

## 3.2 Estrutura da RNN

Para fazer a previsão foi utilizado o modelo de RNN LSTM junto com a técnica da janela.

### 3.2.1 Técnica da janela

Para entender a técnica da janela, e como ela será aplicada no presente trabalho, suponha inicialmente aplicar a técnica a uma sequência genérica  $\{g_1, g_2, g_3, g_4, \dots, g_{n-1}, g_n\}$ , onde se queira prever o próximo termo da sequência usando apenas o termo anterior, ou seja, a janela terá tamanho 1. Assim temos:

Entrada		Previsão
$g_1$	$\longrightarrow$	$g_2$
$g_2$	$\longrightarrow$	$g_3$
$g_3$	$\longrightarrow$	$g_4$
$g_4$	$\longrightarrow$	$g_5$
$\vdots$	$\vdots$	$\vdots$
$g_{n-2}$	$\longrightarrow$	$g_{n-1}$
$g_{n-1}$	$\longrightarrow$	$g_n$

Observe que a previsão começa deslocada, começa em  $g_2$ , isso ocorre porque a janela é de tamanho 1, ou seja, para prever um valor da sequência é necessário o valor anterior ao

valor que se quer prever. Portanto, não há como prever o valor  $g_1$ , pois seria necessário um valor anterior a  $g_1$ .

Agora suponha que o se queira prever o próximo termo da sequência, entretanto, usando uma janela de tamanho 3, esse será o tamanho da janela no presente trabalho. Assim temos que a entrada agora é um vetor de tamanho 3 como segue:

Entrada		Previsão
$\begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}$	$\longrightarrow$	$g_4$
$\begin{bmatrix} g_2 \\ g_3 \\ g_4 \end{bmatrix}$	$\longrightarrow$	$g_5$
$\begin{bmatrix} g_3 \\ g_4 \\ g_5 \end{bmatrix}$	$\longrightarrow$	$g_6$
$\vdots$	$\vdots$	$\vdots$
$\begin{bmatrix} g_{n-4} \\ g_{n-3} \\ g_{n-2} \end{bmatrix}$	$\longrightarrow$	$g_{n-1}$
$\begin{bmatrix} g_{n-3} \\ g_{n-2} \\ g_{n-1} \end{bmatrix}$	$\longrightarrow$	$g_n$

Observe que como no exemplo anterior, agora a previsão também começa deslocada, começa em  $g_4$ , isso ocorre porque a janela é de tamanho 3, ou seja, para prever um valor da sequência são necessários os 3 valores anteriores ao valor que se quer prever. Portanto, não há como prever os valores  $g_1, g_2$  e  $g_3$ , pois seriam necessários três valores anteriores a  $g_1$ , o mesmo se aplica a  $g_2$  e  $g_3$ .

No presente trabalho em vez de uma sequência tem-se 5 sequências como segue:

$$x_{1_{Temperature}}, x_{2_{Temperature}}, x_{3_{Temperature}}, x_{4_{Temperature}}, \dots, x_{n_{Temperature}}$$

$$x_{1_{Humidity}}, x_{2_{Humidity}}, x_{3_{Humidity}}, x_{4_{Humidity}}, \dots, x_{n_{Humidity}}$$

$$x_{1_{Irradiance}}, x_{2_{Irradiance}}, x_{3_{Irradiance}}, x_{4_{Irradiance}}, \dots, x_{n_{Irradiance}}$$

$$x_{1_{Pressure}}, x_{2_{Pressure}}, x_{3_{Pressure}}, x_{4_{Pressure}}, \dots, x_{n_{Pressure}}$$



$$y_{1_{Power}}, y_{2_{Power}}, y_{3_{Power}}, y_{4_{Power}}, \dots, y_{n_{Power}}$$

A janela que será usada aqui é de tamanho 3. A diferença entre a previsão que será feita no presente trabalho e as dos exemplos anteriores, esta no fato que é a técnica da janela será aplicada a quatro sequências ao mesmo tempo para fazer a previsões da quinta sequência. Assim temos:

Entrada				Previsão
$\begin{bmatrix} x_{1_{Temperature}} \\ x_{2_{Temperature}} \\ x_{3_{Temperature}} \end{bmatrix}$	$\begin{bmatrix} x_{1_{Humidity}} \\ x_{2_{Humidity}} \\ x_{3_{Humidity}} \end{bmatrix}$	$\begin{bmatrix} x_{1_{Irradiance}} \\ x_{2_{Irradiance}} \\ x_{3_{Irradiance}} \end{bmatrix}$	$\begin{bmatrix} x_{1_{Pressure}} \\ x_{2_{Pressure}} \\ x_{3_{Pressure}} \end{bmatrix}$	$\longrightarrow y_{4_{Power}}$
$\begin{bmatrix} x_{2_{Temperature}} \\ x_{3_{Temperature}} \\ x_{4_{Temperature}} \end{bmatrix}$	$\begin{bmatrix} x_{2_{Humidity}} \\ x_{3_{Humidity}} \\ x_{4_{Humidity}} \end{bmatrix}$	$\begin{bmatrix} x_{2_{Irradiance}} \\ x_{3_{Irradiance}} \\ x_{4_{Irradiance}} \end{bmatrix}$	$\begin{bmatrix} x_{2_{Pressure}} \\ x_{3_{Pressure}} \\ x_{4_{Pressure}} \end{bmatrix}$	$\longrightarrow y_{5_{Power}}$
$\begin{bmatrix} x_{3_{Temperature}} \\ x_{4_{Temperature}} \\ x_{5_{Temperature}} \end{bmatrix}$	$\begin{bmatrix} x_{3_{Humidity}} \\ x_{4_{Humidity}} \\ x_{5_{Humidity}} \end{bmatrix}$	$\begin{bmatrix} x_{3_{Irradiance}} \\ x_{4_{Irradiance}} \\ x_{5_{Irradiance}} \end{bmatrix}$	$\begin{bmatrix} x_{3_{Pressure}} \\ x_{4_{Pressure}} \\ x_{5_{Pressure}} \end{bmatrix}$	$\longrightarrow y_{6_{Power}}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\begin{bmatrix} x_{n-4_{Temperature}} \\ x_{n-3_{Temperature}} \\ x_{n-2_{Temperature}} \end{bmatrix}$	$\begin{bmatrix} x_{n-4_{Humidity}} \\ x_{n-3_{Humidity}} \\ x_{n-2_{Humidity}} \end{bmatrix}$	$\begin{bmatrix} x_{n-4_{Irradiance}} \\ x_{n-3_{Irradiance}} \\ x_{n-2_{Irradiance}} \end{bmatrix}$	$\begin{bmatrix} x_{n-4_{Pressure}} \\ x_{n-3_{Pressure}} \\ x_{n-2_{Pressure}} \end{bmatrix}$	$\longrightarrow y_{n-1_{Power}}$
$\begin{bmatrix} x_{n-3_{Temperature}} \\ x_{n-2_{Temperature}} \\ x_{n-1_{Temperature}} \end{bmatrix}$	$\begin{bmatrix} x_{n-3_{Humidity}} \\ x_{n-2_{Humidity}} \\ x_{n-1_{Humidity}} \end{bmatrix}$	$\begin{bmatrix} x_{n-3_{Irradiance}} \\ x_{n-2_{Irradiance}} \\ x_{n-1_{Irradiance}} \end{bmatrix}$	$\begin{bmatrix} x_{n-3_{Pressure}} \\ x_{n-2_{Pressure}} \\ x_{n-1_{Pressure}} \end{bmatrix}$	$\longrightarrow y_{n_{Power}}$

### 3.2.2 Otimizador e função de perda (*loss function*)

Como mencionado anteriormente, o trabalho do otimizador é implementar o algoritmo de *Backpropagation*. Há diversos otimizadores, Chollet et al. (2015b) lista alguns deles:

- *Stochastic gradient descent optimizer - SGD*;
- *Adagrad*;
- *Adadelata*;
- *Adamax*;
- *Adam*.

No presente trabalho optou-se por usar o otimizador *adam*. Adam é um algoritmo para otimização de primeira ordem de funções objetivas estocásticas, baseado em estimativas adaptativas de momentos de ordem inferior. O método é simples de implementar, é computacionalmente eficiente, possui poucos requisitos de memória, é invariável ao redimensionamento diagonal dos gradientes, e é adequado para problemas grandes em termos de dados e/ou parâmetros (KINGMA; BA, 2014).

Assim como otimizadores há também segundo Chollet et al. (2015a) muitas funções de perda (*loss functions*), pode-se citar por exemplo:

- *Mean-Squared Error*;
- *Mean-Absolute Error*;
- *Mean-Squared Logarithmic Error*;
- *Categorical Crossentropy*.

No presente adotou-se *Mean-Squared Error* (*mse*), mostrada na equação 3.1, como função de perda.

$$mse = \frac{1}{N} \sum_{i=1}^N (y'_i - y_i)^2 \quad (3.1)$$

Onde  $y'_i$  é um valor previsto,  $y_i$  é um valor real,  $N$  é o número de valores previstos, os valores reais e previstos são em mesmo número.

### 3.2.3 Cálculo de erro

O erro foi calculado usando a raiz quadrada do erro quadrático médio (*Root-Mean-Square Error - RMSE*):

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (y'_i - y_i)^2} \quad (3.2)$$

onde  $y'$  é o valor predito e  $y$  é o valor real.

### 3.2.4 Normalização dos dados

Segundo Silva (2016), há a necessidade de pré-processamento das amostras de treinamento e teste visando aspectos de melhoria do desempenho de treinamento. Isso implica escalonar as respectivas amostras para a faixa de variação dinâmica das funções de ativação das camadas escondidas. Os dados foram normalizados utilizando a equação 3.3.

$$x' = \frac{x - \min(\text{conjunto de dados})}{\max(\text{conjunto de dados}) - \min(\text{conjunto de dados})} \quad (3.3)$$

Após a realização das simulações, os dados de saída da rede neuronal são reconvertidos para a escala real de acordo com a equação 3.4.

$$x = x' \cdot (\max(\text{conjunto de dados}) - \min(\text{conjunto de dados})) + \min(\text{conjunto de dados}) \quad (3.4)$$

Onde  $x$  é o valor a ser normalizado e  $x'$  é o valor normalizado. No presente trabalho os dados foram normalizado no intervalo 0 a 1, a escolha desse intervalo foi para evitar o problema dos gradientes explosivos (*exploding gradients*). Segundo Brownlee (2017), os gradientes explosivos são um problema no qual grandes gradientes de erro se acumulam e resultam em atualizações muito grandes dos pesos dos modelos de redes neurais durante o treinamento. Isso faz com que o modelo seja instável e incapaz de aprender com os dados de treinamento.

### 3.2.5 Época de treinamento

Durante o processo de treinamento de redes neurais artificiais, cada apresentação completa das amostras pertencentes ao subconjunto de treinamento, visando, sobretudo, o ajuste dos pesos sinápticos e limiares de seus neurônios, será denominado de época de treinamento (SILVA, 2016).

### 3.2.6 Tamanho do lote *Batch size*

Segundo Academy (2019), o tamanho do lote (*Batch Size*) é um termo usado em aprendizado de máquina e refere-se ao número de exemplos de treinamento usados em uma iteração. O *batch size* pode ser uma das três opções:

- ***batch mode***: onde o tamanho do lote é igual ao conjunto de dados total, tornando os valores de iteração e épocas equivalentes;
- ***mini-batch mode***: onde o tamanho do lote é maior que um, mas menor que o tamanho total do conjunto de dados. Geralmente, um número que pode ser dividido no tamanho total do conjunto de dados;
- ***stochastic mode***: onde o tamanho do lote é igual a um. Portanto, o gradiente e os parâmetros da rede neural são atualizados após cada amostra.

### 3.2.7 Modelo adotado

Foram implementados três modelos de rede LSTM. Cada rede fazendo uso da técnica da janela, usando como otimizador *Adam* (KINGMA; BA, 2014), que mostrou eficiência nos testes realizados. A função de perdas, ou *loss*, adotada foi o Erro Quadrático Médio (*Mean-Squared Error*). As redes foram treinadas para 10, 50 e 100 épocas com um

*batch size* de 16. A escolha dos três modelos apresentados aqui, deu-se por tentativa e erro. Foram implementados vários modelos diferentes que culminaram na escolha dos três modelos apresentados a seguir:

- *model1*: modelo com camada de entrada com 256 células (ou blocos) LSTM com função de ativação *ReLU*, camada de saída com 1 neurônio com função de ativação *logistic sigmoid*, otimizador ADAM e função de perda (*loss function*) MSE (Mean Squared Error).
- *model2*: modelo com camada de entrada com 256 células (ou blocos) com função de ativação *ReLU*. Uma camada oculta com 256 células (ou blocos) com função de ativação *ReLU* e camada de saída com 1 neurônio com função de ativação *logistic sigmoid*, otimizador ADAM e função de perda (*loss function*) MSE (Mean Squared Error).
- *model3*: modelo com camada de entrada com 256 blocos LSTM com função de ativação *ReLU*. Três camadas ocultas uma com 256 e duas com 128 células (ou blocos) LSTM com função de ativação *ReLU* e camada de saída com 1 neurônio com função de ativação *logistic sigmoid*, otimizador ADAM e função de perda (*loss function*) MSE (Mean Squared Error).

### 3.2.8 Programação

A rede neural LSTM, foi modelada usando linguagem de programação *Python* em sua versão 3.7, rodando a plataforma *TensorFlow* em sua versão 1.7, junto com a interface de programação de aplicativo (*Application Programming Interface* - API) *Keras*. *Keras* é uma API desenvolvida com o objetivo de permitir experimentação rápida, ser capaz de passar da ideia para o resultado com o menor atraso possível. O programa foi compilado e executado na plataforma *Google Colaboratory*.

## 4 Testes e análise de resultados

Foram testados todos os modelos com os três conjuntos de dados (*data1*, *data2*, *data3*). O melhor modelo para cada conjunto de dados foi obtido a partir da comparação dos erros de cada modelo para as várias épocas consideradas (10, 50 e 100). Os tempos de treinamento dos modelos com os melhores resultados estão na Tabela B.1

### 4.1 Conjunto de dados *data1* (15 minutos)

Os três modelos foram treinados com o conjunto de dados *data1*, algumas características estatísticas deste conjunto de dados podem ser encontradas na Figura A.1 do Anexo A . Com os modelos treinados fez-se previsões com o conjunto de teste. Os dados da Tabela 4.1 são os erros, em kWh, de previsão de cada modelo em relação a todo o conjunto de teste. Nota-se que o melhor modelo foi o *model3*, apresentando  $RMSE = 2,0630329$  kWh, para o conjunto de teste, para 50 épocas de treinamento.

Tabela 4.1 – Valores de erros em kWh, conjunto *data1*.

	<i>data1</i>		
Modelo/Época	10	50	100
<i>model1</i>	2,29014020	2,1558444	2,1706862
<i>model2</i>	2,22402020	2,1018310	2,1713316
<i>model3</i>	2,11206360	<b>2,0630329</b>	2,1070063

Fonte: O autor.

Este modelo apresentou um erro de aproximadamente 9,421% em relação ao maior valor do conjunto de teste.

### 4.2 Conjunto de dados *data2* (60 minutos)

Os três modelos foram treinados com o conjunto de dados *data2*, algumas características estatísticas deste conjunto de dados podem ser encontradas na Figura A.2 do Anexo A. Com os modelos treinados fez-se previsões com o conjunto de teste. O dado da Tabela 4.2 são os erros, em kWh, da previsão de cada modelo em relação a todo o conjunto de teste. Nota-se que o melhor modelo foi o *model3*, apresentando  $RMSE = 1,26262440$  kWh, para o conjunto de teste, para 100 épocas de treinamento.

O erro encontrado com esse modelo foi o menor erro para todos os conjuntos de dados e modelos, este resultado já era esperado, devido a correlação entre a *Irradiance*

Tabela 4.2 – Valores de erros em kWh, conjunto *data2*.

<i>data2</i>			
Modelo/Época	10	50	100
<i>model1</i>	1,56816960	1,87383850	1,63655150
<i>model2</i>	1,43026360	1,58151090	1,48509570
<i>model3</i>	1,63364120	1,56337240	<b>1,26262440</b>

Fonte: O autor.

e *Power* para o conjunto *data2*, amostragem em horas, ser de 0,99 positiva. Contudo não esperava-se que um modelo relativamente simples como esse com poucas épocas de treinamento demonstrasse tamanha eficiência. Este modelo apresentou um erro de aproximadamente 6,189% em relação ao maior valor do conjunto de teste.

### 4.3 Conjunto de dados *data3* (720 minutos)

Os três modelos foram treinados com o conjunto de dados *data3*, algumas características estatísticas deste conjunto de dados podem ser encontradas na Figura A.3 do Anexo A. Com os modelos treinados fez-se previsões com o conjunto de teste. O dado da Tabela 4.2 são os erros, em kWh, da previsão de cada modelo em relação a todo o conjunto de teste. Nota-se que o melhor modelo foi o *model1*, apresentando RMSE = 12,403618 kWh, para o conjunto de teste, para 100 épocas de treinamento.

Tabela 4.3 – Valores de erros em kWh, conjunto *data3*.

<i>data3</i>			
Modelo/Época	10	50	100
<i>model1</i>	21,4275860	17,990059	<b>12,403618</b>
<i>model2</i>	21,5310330	18,381023	16,366114
<i>model3</i>	21,9092240	20,294960	19,622486

Fonte: O autor.

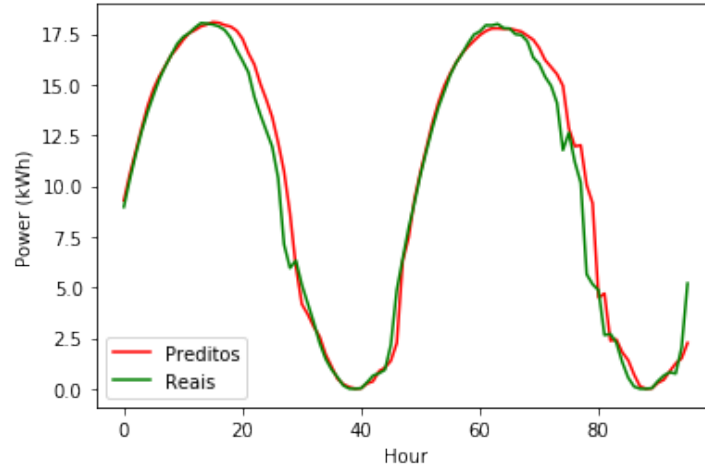
Este modelo apresentou um erro de aproximadamente 7,988% em relação ao maior valor do conjunto de teste.

### 4.4 Resultados gráficos

Foram plotados para cada modelo e conjunto de dados um gráfico de previsão do modelo com menor erro, indicando os valores previstos e os valores reais. A seguir pode-se observar que os gráficos quase se sobrepõem. O gráfico da Figura 4.1 corresponde ao do *model3* para o conjunto de dados *data1*, este foi o modelo que teve melhor desempenho segundo a Tabela 4.1. Para este gráfico foram utilizados as 96 primeiras amostras do

conjunto de teste o que equivale a dois dias de medições de 15 em 15 minutos das 6 as 18 horas.

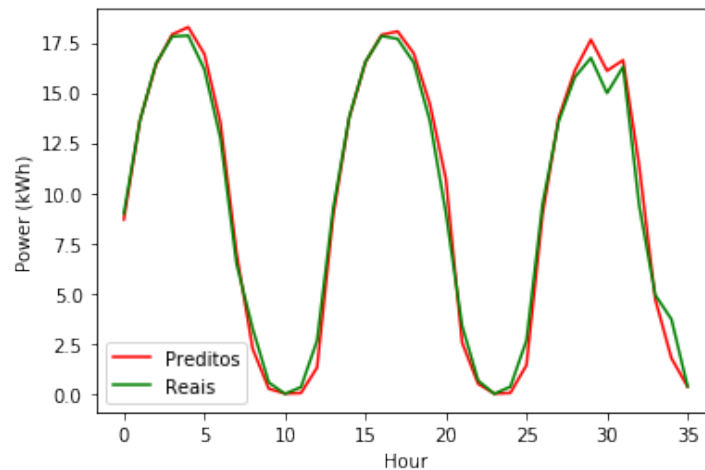
Figura 4.1 – Gráfico *model3*, previsão de 96 pontos



Fonte: O autor.

O gráfico da Figura 4.2 é gráfico do *model3*, para o conjunto de dados *data2*, este

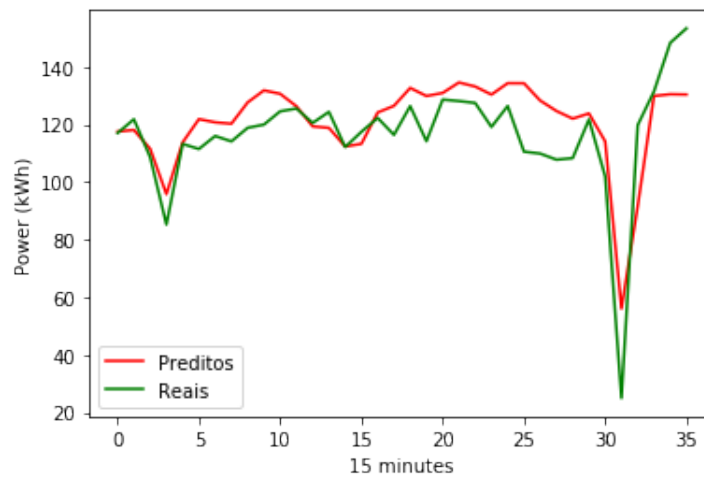
Figura 4.2 – Gráfico *model3*, previsão de 36 pontos



Fonte: O autor.

foi o modelo que teve melhor desempenho segundo a Tabela 4.2. Para este gráfico foram utilizados as 36 primeiras amostras do conjunto de teste, o que equivale a três dias de medições a cada hora no intervalo das 6 as 18 horas. O gráfico da Figura 4.3, do *model1* para o conjunto de dados *data3*, apresenta 36 dias de previsão dia a dia, apenas no intervalo de 6 às 18 horas.

Pode-se notar do gráfico, que já não temos uma boa aproximação em relação aos valores reais. Um dos motivos pode ser o uso de médias das variáveis climáticas como

Figura 4.3 – Gráfico *model1*, previsão de 36 pontos

Fonte: O autor.

entrada para um intervalo de tempo tão grande.

Observa-se que em uma comparação geral entre os modelos, o modelo 3 se sobrepõe aos outros modelos, apresentado erros menores, nos horizontes considerados a muito curto prazo.



## 5 Conclusão

No âmbito deste trabalho foi proposta e apresentada uma metodologia de previsão de geração de uma usina fotovoltaica a muito curto prazo e curto prazo. Intencionou-se assim, culminar no desenvolvimento de uma ferramenta computacional robusta e eficiente, capaz de viabilizar a solução de problemas associados com a análise da operação da rede, a qual a usina está conectada, sob condições de incerteza.

A metodologia consistiu no uso de dados climáticos e redes neurais recorrentes para fazer as previsões. Foram adotados três horizontes de tempos distintos e propostos três modelos de redes neurais recorrentes. O modelo de maior complexidade, o *model3*, com uma camada de entrada, cinco camadas ocultas e uma camada de saída, destacou-se dos outros, apresentando desvios relativamente menores, considerando os horizontes de quinze minutos e de uma hora, horizontes de muito curto prazo. Os gráficos gerados a partir das previsões desse modelo praticamente sobrepuaram-se aos gráficos obtidos com amostras reais. No horizonte de curto prazo houve um modelo que destacou-se dos outros, o *model1*, o mais simples de todos, com apenas uma camada de entrada e uma de saída. Esse fato, parece indicar uma possível relação inversa entre a complexidade do modelo e uma redução nos desvios das previsões para horizontes de tempo maiores.

Conclui-se que o trabalho desenvolvido atingiu os objetivos propostos no plano de trabalho. Como mostrado nos resultados, os modelos propostos para a previsão da produção da usina, considerando os horizontes de análise, obtiveram previsões razoáveis. Ainda baseando-se no desvio das previsões obtidas e na complexidade dos modelos, pode-se concluir que há a necessidade de mais estudos que apoiem ou refutem uma relação entre a alta (ou baixa) complexidade do modelo adotado, e a qualidade das previsões para horizontes de tempo maiores (ou menores).

### 5.1 Trabalhos futuros

A validade do método computacional com redes neurais foi demonstrada, como propunha o presente trabalho. Contudo este trabalho é apenas um tímido vislumbre das possibilidades a frente. Fica aqui sugestões para melhoria do método e trabalhos futuros:

- Aumentar a quantidade de amostras e obter uma relação entre o ajuste, erro médio, e a quantidade amostrada;
- Confirmar ou refutar a relação entre a alta (ou baixa) complexidade do modelo adotado, e a qualidade das previsões para horizontes de tempo maiores (ou menores);

- Verificar a eficiência e robustez dos modelos utilizando como dados de entrada apenas os valores geração da usina;
- Utilizar métodos de *boosting*, que servem para otimizar os modelos;
- Aplicar os modelos propostos a outros tipos de fontes renováveis;
- Estabelecer um modelo de previsão para a usina que possa ser automaticamente ajustada a cada nova amostra obtida;
- Propor modelos de simulação que identifiquem o nível de distorção harmônica na saída da usina.

# Referências

ABDEL-NASSER, M.; MAHMOUD, K. Accurate photovoltaic power forecasting models using deep lstm-rnn. *Neural Comput and Applic (2019) 31: 2727*, p. 2727 - 274, 2017. Citado 3 vezes nas páginas 24, 34 e 35.

ACADEMY, D. S. *Deep Learning Book*. 2019. Disponível em: <<http://deeplearningbook.com.br/o-efeito-do-batch-size-no-treinamento-de-redes-neurais-artificiais/>>. Acesso em: 08 dez. 2019. Citado na página 49.

ANALYZER, T. *Theia Analyzer*. 2019. Disponível em: <<http://200.137.220.91/INDEX.HTM>>. Acesso em: 01 nov. 2019. Citado na página 42.

ANDRADE, L. C. M. de. *Abordagem Neurofuzzy Para Previsão de Demanda de Energia Elétrica no Curtíssimo Prazo*. Dissertação (Mestrado) — Escola de Engenharia Elétrica de São Carlos, USP, São Carlos, SP, mar. 2010. Citado na página 24.

ANEEL. *Micro e Minigeração Distribuída: Sistema de Compensação de Energia Elétrica*. 2. ed. Brasília: Agência Nacional de Energia Elétrica, 2016. Acesso em: 1 ago 2019. Citado na página 21.

ANEEL. *Revisão das regras aplicáveis à micro e minigeração distribuída – Resolução Normativa nº 482/2012 - Relatório de Análise de Impacto Regulatório nº 0004/2018-SRD/SCG/SMA/ANEEL*. [S.l.], 2018. Disponível em: <[aneel.gov.br/documents/656877/18485189/6+Modelo+de+AIR+-+SRD+-+Geraç~ao+Distribuida.pdf/769daa1c-51af-65e8-e4cf-24eba4f965c1](http://aneel.gov.br/documents/656877/18485189/6+Modelo+de+AIR+-+SRD+-+Geraç~ao+Distribuida.pdf/769daa1c-51af-65e8-e4cf-24eba4f965c1)>. Acesso em: 29 set. 2019. Citado na página 22.

ANEEL. *O que o orgulho dá - Passado, presente e futuro da energia gerada em casa*. Brasília: Agência Nacional de Energia Elétrica, 2019. Disponível em: <<https://www.aneel.gov.br/documents/656877/19367596/01.png/686e30c8-892e-e871-daa7-05f251827876?t=1571157566720>>. Acesso em: 5 dez. 2019. Citado na página 22.

BROWNLEE, J. *Long Short-Term Memory Networks With Python: Develop Sequence Prediction Models With Deep Learning*. Melbourne, AUS: Machine Learning Mastery, 2017. Citado 2 vezes nas páginas 34 e 49.

CEPEL-CRESEB. *Manual de Engenharia para Sistemas Fotovoltaicos*. [S.l.], 2014. Disponível em: <[http://www.cresesb.cepel.br/publicacoes/download/Manual\\_de\\_Engenharia\\_FV\\_2014.pdf](http://www.cresesb.cepel.br/publicacoes/download/Manual_de_Engenharia_FV_2014.pdf)>. Acesso em: 11 ago. 2019. Citado na página 21.

CHOLLET, F. *Deep Learning with Python*. Shelter Island, NY, USA.: Manning, 2018. Citado 2 vezes nas páginas 32 e 33.

CHOLLET, F. et al. *Usage of loss functions*. 2015. Disponível em: <<https://keras.io/losses/#usage-of-loss-functions>>. Acesso em: 01 ago 2019. Citado na página 48.

CHOLLET, F. et al. *Usage of optimizers*. 2015. Disponível em: <<https://keras.io/optimizers/>>. Acesso em: 01 ago. 2019. Citado na página 47.

CIABURRO, G.; VENKATESWARAN, B. *Neural Networks with R: Smart models using CNN, RNN, deep learning, and artificial intelligence principles*. 1. ed. BIRMINGHAM, U.K.: Packt, 2017. Citado 2 vezes nas páginas 26 e 32.

FAUSETT, L. V. *Fundamentals of Neural Networks: Architectures, Algorithms And Applications*. 1. ed. Lebanon, Indiana, USA: Pearson, 1993. Citado na página 28.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, MA: MIT Press, 2017. Citado na página 31.

HEBB, D. O. *The Organization of Behavior: A Neuropsychological Theory*. New York, USA: Wiley, 1949. Citado na página 25.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. Citado 2 vezes nas páginas 48 e 49.

LOPES, J. H. O. *Previsão de produção fotovoltaica a muito curto prazo usando análise e modelação de dinâmica de sombras*. Dissertação (Mestrado) — Faculdade de Engenharia da Universidade do Porto, Porto, Jun. 2010. Citado na página 23.

MARRA, E. G.; PIMENTEL, S. P. *Weather Station*. 2015. Disponível em: <<https://sites.google.com/site/sfvemcufg/weather-station>>. Acesso em: 01 nov. 2019. Citado na página 41.

MARRA, E. G.; PIMENTEL, S. P. *2019\_03 EMC-UFG\_Drive*. 2019. Disponível em: <[https://docs.google.com/spreadsheets/d/1iGVeFNfHTMJFiEu8JjLEhP4-gByuwbCZkNr8\\_TkrIF4/edit#gid=2124979687](https://docs.google.com/spreadsheets/d/1iGVeFNfHTMJFiEu8JjLEhP4-gByuwbCZkNr8_TkrIF4/edit#gid=2124979687)>. Acesso em: 09 nov. 2019. Citado na página 42.

MCCULLOCH, W. S.; PITTS, W. H. A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, v. 5, p. 115-113, 1943. Citado 3 vezes nas páginas 25, 26 e 28.

NASCIMENTO, R. L. *Energia solar no Brasil: situação e perspectivas*. [S.l.], 2017. Disponível em: <[http://bd.camara.gov.br/bd/bitstream/handle/bdcamara/32259/energia\\_solar\\_limp.pdf?sequence=1](http://bd.camara.gov.br/bd/bitstream/handle/bdcamara/32259/energia_solar_limp.pdf?sequence=1)>. Acesso em: 11 ago. 2019. Citado na página 21.

OLAH, C. *Understanding LSTM Networks*. 2015. Disponível em: <<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>>. Acesso em: 08 dez. 2019. Citado 4 vezes nas páginas 35, 36, 37 e 38.

SANTOS, B. A. de O. *Previsão de Produção de Centrais Solares a Muito Curto Prazo*. Dissertação (Mestrado) — Faculdade de Engenharia da Universidade do Porto, Porto, Jun. 2010. Citado na página 23.

SILVA, I. N. da. *Redes neurais artificiais: para engenharia e ciências aplicadas*. 2. ed. São Paulo: Person, 2016. Citado 8 vezes nas páginas 25, 26, 27, 28, 29, 30, 48 e 49.

SU, D.; BATZELIS, E.; PAL, B. Machine learning algorithms in forecasting of photovoltaic power generation. *IEEE*, 2019. Citado 2 vezes nas páginas 24 e 25.

## Anexos



## ANEXO A – Principais informações estatísticas dos conjuntos de dados trabalhados

Figura A.1 – Principais características estatísticas do conjunto *data1*.

	Temperature	Humidity	Irradiance	Pressure	Power
<b>count</b>	8771.000000	8771.000000	8771.000000	8771.000000	8771.000000
<b>mean</b>	26.717405	50.044872	421.393306	943.027556	8.935951
<b>std</b>	4.932906	20.392774	298.776456	3.018549	6.546944
<b>min</b>	8.142667	12.464000	0.000000	934.233333	0.000000
<b>25%</b>	23.231667	34.354667	125.390000	940.861167	2.122500
<b>50%</b>	27.356000	47.068000	425.800000	943.173400	9.294000
<b>75%</b>	30.140667	63.452000	696.583333	945.155993	15.199500
<b>max</b>	39.016000	99.990000	1128.971933	951.615167	24.431000

Fonte: O autor.

Figura A.2 – Principais características estatísticas do conjunto *data2*.

	Temperature	Humidity	Irradiance	Pressure	Power
<b>count</b>	2327.000000	2327.000000	2327.000000	2327.000000	2327.000000
<b>mean</b>	26.288157	51.426780	397.108646	942.912688	8.427563
<b>std</b>	5.097765	20.911216	297.519009	3.017693	6.542773
<b>min</b>	8.211167	13.317667	0.000000	934.174833	0.000000
<b>25%</b>	22.313917	35.221500	93.407792	940.760758	1.578833
<b>50%</b>	26.996667	48.420833	402.118333	942.997485	8.847333
<b>75%</b>	29.964250	65.760083	671.246601	945.016596	14.695833
<b>max</b>	38.038000	99.990000	1044.537115	951.583670	22.003000

Fonte: O autor.

Figura A.3 – Principais características estatísticas do conjunto *data3*.

	Temperature	Humidity	Irradiance	Pressure	Power
<b>count</b>	181.000000	181.000000	181.000000	181.000000	181.000000
<b>mean</b>	24.666165	56.388374	286.768559	942.495847	109.466790
<b>std</b>	2.586959	15.755653	45.962615	2.558409	15.060662
<b>min</b>	15.796037	22.495787	56.431586	935.816639	24.938999
<b>25%</b>	22.747093	44.541907	268.388556	941.010860	105.356003
<b>50%</b>	24.592611	56.564796	288.466578	942.558585	111.411003
<b>75%</b>	26.141130	66.969472	313.015677	944.368296	117.630997
<b>max</b>	30.828731	90.155833	390.393563	948.531023	153.354996

Fonte: O autor.



## ANEXO B – Tempos de treinamento

Tabela B.1 – Tempos de treinamento dos modelos com melhores resultados.

<i>Model/Data</i>	Tempo de treinamento (min)
<i>model3/data1</i>	11,80447
<i>model3/data2</i>	6,31474
<i>model1/data3</i>	0,16707

Fonte: O autor.