



UNIVERSIDADE FEDERAL DE SANTA CATARINA
CENTRO TECNOLÓGICO
DEPARTAMENTO DE ENGENHARIA ELÉTRICA E ELETRÔNICA

Predição de Geração Fotovoltaica Usando Aprendizado Profundo

Stevan Sehn

Orientador: Eduardo Luiz Ortiz Batista
Coorientador: Walter Antônio Gontijo

Florianópolis

2021

Ficha de identificação da obra elaborada pelo autor,
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Sehn, Stevan

Predição de Geração Fotovoltaica Usando Aprendizado
Profundo / Stevan Sehn ; orientador, Eduardo Luiz Ortiz
Batista, coorientador, Walter Antônio Gontijo, 2021.
75 p.

Trabalho de Conclusão de Curso (graduação) -
Universidade Federal de Santa Catarina, Centro Tecnológico,
Graduação em Engenharia Elétrica, Florianópolis, 2021.

Inclui referências.

1. Engenharia Elétrica. 2. Energia Solar Fotovoltaica.
3. Aprendizado de Máquina. I. Luiz Ortiz Batista, Eduardo
. II. Antônio Gontijo, Walter. III. Universidade Federal de
Santa Catarina. Graduação em Engenharia Elétrica. IV. Título.

Predição de Geração Fotovoltaica Usando Aprendizado Profundo

Trabalho de Conclusão de Curso submetido ao curso de Engenharia Elétrica da Universidade Federal de Santa Catarina como requisito para aprovação da disciplina EEL7890 - Trabalho de Conclusão de Curso (TCC).

Orientador: Eduardo Luiz Ortiz Batista
Coorientador: Walter Antônio Gontijo

Florianópolis
2021

Stevan Sehn

Predição de Geração Fotovoltaica Usando Aprendizado Profundo

Este Trabalho de Conclusão de Curso foi julgado adequado para obtenção do Título de “Bacharel em Engenharia Elétrica” e aceito, em sua forma final, pelo Curso de Graduação em Engenharia Elétrica.

Florianópolis, 31 de março de 2021.



Documento assinado digitalmente
Jean Viane Leite
Data: 19/04/2021 10:48:56-0300
CPF: 003.474.909-80
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Jean Viane Leite, Dr.
Coordenador do Curso de Graduação em Engenharia Elétrica

Banca Examinadora:



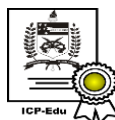
Documento assinado digitalmente
Eduardo Luiz Ortiz Batista
Data: 11/04/2021 00:43:51-0300
CPF: 036.521.889-85
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Eduardo Luiz Ortiz Batista, Dr.
Orientador
Universidade Federal de Santa Catarina



Documento assinado digitalmente
Walter Antonio Gontijo
Data: 12/04/2021 09:06:23-0300
CPF: 536.287.996-00
Verifique as assinaturas em <https://v.ufsc.br>

Prof. Walter Antônio Gontijo, M.Sc.
Coorientador
Universidade Federal de Santa Catarina



Documento assinado digitalmente
Helena Flavia Naspolini
Data: 19/04/2021 16:08:39-0300
CPF: 289.588.109-04
Verifique as assinaturas em <https://v.ufsc.br>

Profª. Helena Flávia Naspolini, Dr.
Universidade Federal de Santa Catarina

Eng. Eletric. Natan Votre
Brick Abode

Dedico este TCC a Anthony Francis Glenn (*in memoriam*).

AGRADECIMENTOS

Agradeço à minha família pelo apoio, à UFSC e ao LINSE pela formação, e ao Walter pelo direcionamento.

*Quando vedes a nuvem que vem do ocidente, logo
dizeis: Lá vem chuva, e assim sucede.*

Lucas 12:54

RESUMO

A previsão confiável da geração de energia fotovoltaica é um fito desejável para operadores do sistema elétrico, uma vez que a integração desta e outras fontes renováveis à rede traz desafios para a adequação entre oferta e demanda de energia. O foco deste trabalho está na predição da potência gerada por um sistema fotovoltaico utilizando aprendizado de máquina. Para tal, utiliza-se um conjunto de dados relacionando dados ambientais de Florianópolis com dados de geração de um sistema fotovoltaico situado no mesmo município. Os resultados obtidos demonstram o potencial do uso de aprendizado de máquina para a aplicação considerada.

Palavras-chave: Energia Solar Fotovoltaica, Aprendizado de Máquina.

ABSTRACT

The reliable prediction of photovoltaic power generation is a desirable goal for power system operators, since the integration of this and other renewable sources into the power grid brings challenges to the task of matching power supply and demand. The focus of this work is on the prediction of the generated power of a photovoltaic system utilizing machine learning. To this end, a dataset that relates environmental data from Florianopolis to power generation data of a photovoltaic system in the city is used. The results obtained demonstrate the potential of using machine learning for the considered application.

Keywords: Solar Photovoltaic Power, Machine Learning.

Sumário

1 INTRODUÇÃO	1
1.1 OBJETIVOS	2
1.1.1 Objetivo Geral	2
1.1.2 Objetivos Específicos	2
1.2 METODOLOGIA	2
1.2.1 Pesquisa Bibliográfica	3
1.2.1.1 Revisão Bibliográfica Geral	3
1.2.1.2 Revisão de Literatura sobre Aplicação de Aprendizado de Máquina na Previsão de Geração Fotovoltaica	3
1.2.2 Pesquisa Experimental	3
1.2.2.1 Definição do Problema	4
2 FUNDAMENTAÇÃO TEÓRICA	4
2.1 ENERGIA SOLAR	5
2.1.1 O Recurso Solar	5
2.2 SISTEMAS FOTOVOLTAICOS	10
2.2.1 A Célula Fotovoltaica	10
2.2.2 O Módulo Fotovoltaico	15
2.3 APRENDIZADO DE MÁQUINA	18
2.3.1 Visão Geral de Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo	18
2.3.2 Aprendizado de Máquina	20
2.3.2.1 Regressão Linear	21
2.3.2.2 Regressão Logística	24
2.3.2.3 Otimização	26
2.3.2.4 Overfitting	27
2.3.3 Aprendizado Profundo	27
2.3.3.1 Rede Neural Feedforward	28
2.3.3.2 Outros Métodos de Otimização	32
2.4 A SEQUÊNCIA DE PROCEDIMENTOS DE APRENDIZADO DE MÁQUINA	34
2.4.1 Definição do Problema de Aprendizado de Máquina	34
2.4.2 Escolha do Protocolo de Validação	35
2.4.3 Pré-Processamento e Engenharia de Atributos	36
2.4.3.1 Pré-Processamento	36
2.4.3.2 Engenharia de Atributos	36
2.4.4 Redução de Overfitting	37

2.4.4.1	Ajuste de Hiperparâmetros	38
2.4.4.2	Regularização	38
2.5	REVISÃO DO ESTADO DA ARTE	39
2.5.1	Discussão sobre os Artigos Pesquisados	41
3	DESENVOLVIMENTO	41
3.1	O CONJUNTO DE DADOS	42
3.1.1	Rede SONDA	42
3.1.2	SUNNY Portal	44
3.1.3	Discussão sobre o Conjunto de Dados	45
3.1.4	Perfil do Sistema Sede	48
3.2	IMPLEMENTAÇÃO	49
3.2.1	Definição do Problema	49
3.2.2	Validação	50
3.2.3	Pré-Processamento e Feature Engineering	50
3.2.4	Redução de Overfitting	53
4	EXPERIMENTOS E RESULTADOS	53
4.1	MODELAGEM	54
4.2	PREVISÃO SIMPLES	64
4.3	PREVISÃO COM O CONJUNTO DE DADOS REESTRUTURADO	68
5	CONSIDERAÇÕES FINAIS	71
5.1	PROPOSTAS PARA TRABALHOS FUTUROS	72

Lista de Figuras

Figura 1:	Órbita da Terra em torno do Sol	6
Figura 2:	Varição da irradiância extraterrestre efetiva ao longo do ano . . .	7
Figura 3:	Componentes da radiação solar	8
Figura 4:	Irradiação Normal Global – Mundo	9
Figura 5:	Irradiação Normal Global – Brasil	9
Figura 6:	Estrutura básica de uma célula fotovoltaica de silício	10
Figura 7:	Circuito equivalente básico para uma célula fotovoltaica	12
Figura 8:	Curvas I-V de duas células fotovoltaicas ligadas em a) série e b) paralelo	13
Figura 9:	Influência da variação da irradiância solar na curva característica I-V de uma célula fotovoltaica de silício cristalino na temperatura de 25 °C	14
Figura 10:	Influência da temperatura da célula fotovoltaica na curva I-V (para irradiância de 1000 W/m^2)	15
Figura 11:	Curva característica I-V e curva de potência P-V para um módulo com potência nominal de 100 Wp	16
Figura 12:	Efeito da variação da irradiância solar sobre a curva característica I-V para um módulo fotovoltaico de 36 células de silício cristalino (c-Si) a 25 °C	17
Figura 13:	Efeito da variação da temperatura das células sobre a curva característica I-V para um módulo fotovoltaico de 36 células de silício cristalino (c-Si) sob irradiância de 1000 W/m^2	17
Figura 14:	Relação entre IA, aprendizado de máquina e aprendizado profundo	18
Figura 15:	Diagrama de blocos de uma rede neural	20
Figura 16:	Distâncias entre os pontos de dados e as previsões. O método dos mínimos quadrados visa minimizar a soma dos quadrados destas distâncias	24
Figura 17:	Função Logística	26

Figura 18:	20 ciclos do método do gradiente a taxas de aprendizado a) $\eta = 0.1$ e b) $\eta = 0.6$. O algoritmo começa no ponto (0,0) e o ponto de mínimo global é (1,1)	27
Figura 19:	Rede neural de duas camadas	28
Figura 20:	Redes neurais com uma, duas (uma oculta), três (duas ocultas) e seis (cinco ocultas) camadas	29
Figura 21:	Unidade de Camada ou Neurônio	30
Figura 22:	Funções de ativação	32
Figura 23:	Esquemático do algoritmo de propagação reversa para uma rede de duas camadas	33
Figura 24:	Dados da rede SONDA do ano de 2018	46
Figura 25:	Dados da rede SONDA do ano de 2019	46
Figura 26:	Geração do Sistema “Sede” de 15 de abril de 2018	47
Figura 27:	Geração do Sistema “Sede” de 16 de abril de 2018	47
Figura 28:	Geração do Sistema “Sede” de 5 de dezembro de 2018	47
Figura 29:	Conjunto de dados	51
Figura 30:	Conjunto de dados após feature engineering.	52
Figura 31:	Diagrama de blocos do Modelo A	54
Figura 32:	Diagrama de blocos do Modelo B	55
Figura 33:	Diagrama de blocos do Modelo C	56
Figura 34:	Diagrama em blocos do modelo final	58
Figura 35:	Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e predição (laranja) do modelo final	60
Figura 36:	Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e predição (laranja) do Modelo D	63
Figura 37:	Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e predição (laranja) do modelo final com 30 minutos de antecedência	65
Figura 38:	Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e predição (laranja) do modelo final com 1 hora de antecedência	66
Figura 39:	Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e predição (laranja) do modelo final com 30 minutos de antecedência com o conjunto de dados reestruturado	69
Figura 40:	Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e predição (laranja) do modelo final com 1 hora de antecedência com o conjunto de dados reestruturado	70

1 INTRODUÇÃO

A energia solar é uma fonte de energia limpa, abundante e renovável, e seu uso na produção de energia elétrica, em particular através da geração fotovoltaica, vem crescendo significativamente no Brasil e no mundo. Contudo, a integração de fontes de energia fotovoltaicas às redes de distribuição tradicionais oferece alguns desafios. Isto se deve ao fato de que, ao contrário das fontes tradicionais, a geração fotovoltaica é inconstante – ou “estocástica” [1] –, pois depende, entre outros fatores, das condições climáticas. A energia hidrelétrica também depende de condições climáticas, mas numa escala de tempo bem maior e mais previsível. A geração fotovoltaica, por sua vez, varia diariamente, acarretando problemas para o operador do sistema elétrico, que precisa ter uma previsão da geração para adequá-la à demanda.

Teo *et al.* [2] resumizam o problema da operação conforme descrito a seguir. A operação precisa combinar oferta (ou geração) de energia com a carga (ou demanda). Se a demanda for superior à geração, a frequência da rede cai, enquanto que, se a geração for superior à demanda, a frequência da rede sobe. A estabilidade da rede, portanto, depende da habilidade do operador de manter a frequência constante (em 60 Hz no caso do Brasil). Há tipicamente dois tipos de geradores: geradores de base e de pico (também chamados usinas de carga de base e usinas de energia de pico). Os geradores de base servem para suprir o nível mínimo de demanda em um período. Eles operam 24 horas por dia e demoram mais para entrar em funcionamento. Por esta razão, eles são incapazes de acompanhar as mudanças bruscas de cargas durante o período de pico (ou de ponta) [3]. Os custos de operação com geradores de base são baixos. Exemplos de geradores de base seriam usinas hidrelétricas ou termelétricas a carvão mineral. Já os geradores de pico operam apenas por algumas horas, entram em funcionamento rapidamente e podem ser ajustadas facilmente para satisfazer as variações de potência durante o horário de pico. Contudo, seu custo de operação é mais elevado. Um exemplo de gerador de pico seria uma usina termelétrica a gás natural.

A rede tradicional (atual) é dominada pela carga (*load-dominant*). Contudo, espera-se que, com a integração das fontes de energia renováveis, as redes passem a ser dominadas pela geração (*generation-dominant*). Isto porque as fontes de energia renováveis abrem espaço para pequenos produtores independentes, nas chamadas redes

inteligentes ou *smart grids*. No Brasil, a lei já regulamenta a microgeração e a minigeração distribuídas [4]. Com a entrada destes novos atores, a dinâmica se altera consideravelmente. Pode ocorrer de a geração superar a carga prevista, com os problemas relacionados (aumento da frequência da rede e suas consequências). Neste caso o operador teria que desconectar rapidamente as fontes renováveis da rede, e na medida certa, para prevenir danos. O ideal, no entanto, seria ter-se uma previsão confiável da geração das renováveis para se fazer o planejamento necessário de antemão. É aqui que entra o aprendizado de máquina.

O aprendizado de máquina é um subtópico de inteligência artificial. Pode-se defini-lo como uma série de técnicas destinadas a fazer o computador “aprender” com os dados e fazer previsões baseadas neste aprendizado [5]. Segundo Su *et al.* [1], técnicas de aprendizado de máquina têm se mostrado superiores a métodos estatísticos convencionais na previsão de geração fotovoltaica devido à sua capacidade intrínseca de modelar processos dinâmicos e não lineares.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Implementar um modelo de aprendizado de máquina capaz de prever, com alguma antecedência, a potência gerada em um sistema fotovoltaico usando dados climáticos.

1.1.2 Objetivos Específicos

- Avaliar o comportamento da geração de sistemas fotovoltaicos visando conhecer melhor a natureza do problema proposto e os atributos (*features*) relevantes;
- Obter um dataset que relacione dados ambientais à geração fotovoltaica;
- Treinar e validar modelos de aprendizado de máquina adequados ao problema considerado;
- Realizar ajuste de hiperparâmetros e regulação do modelo.

1.2 METODOLOGIA

A metodologia adotada neste trabalho consiste em uma pesquisa bibliográfica e uma pesquisa experimental. Na pesquisa bibliográfica é feita uma revisão sobre

aprendizado de máquina e energia solar fotovoltaica e uma revisão de literatura a fim de se avaliar o estado da arte. Na pesquisa experimental é feita a aquisição e preparação dos dados e a implementação do modelo de aprendizado de máquina. As etapas são detalhadas a seguir.

1.2.1 Pesquisa Bibliográfica

1.2.1.1 Revisão Bibliográfica Geral

A Revisão Bibliográfica Geral tem por objetivo a fundamentação teórica do trabalho. As fontes pesquisadas são livros, apostilas, artigos, revistas impressas ou eletrônicas, sites, cursos online e outras. A fundamentação teórica é subdividida em dois tópicos: 1) energia solar e sistemas fotovoltaicos e 2) aprendizado de máquina. No tópico sobre energia solar e sistemas fotovoltaicos são abordados: o recurso solar, a geometria Sol-Terra, a radiação solar, tipos de energia solar, unidades e instrumentos de medida (este último é relevante para a definição dos parâmetros que compõem o conjunto de dados utilizado no treinamento, validação e teste do modelo de aprendizado de máquina); o sistema fotovoltaico, princípio de funcionamento (efeito fotoelétrico); componentes básicos: a célula e o módulo fotovoltaicos. Já no tópico sobre aprendizado de máquina são tratados: inteligência artificial em geral (da qual o aprendizado de máquina é uma subdivisão); aprendizado de máquina: conceito, técnicas e abordagens mais utilizadas; aprendizado profundo: redes neurais artificiais, tipos de redes neurais.

1.2.1.2 Revisão de Literatura sobre Aplicação de Aprendizado de Máquina na Previsão de Geração Fotovoltaica

Com esta pesquisa busca-se determinar o estado da arte e ter-se uma visão panorâmica das estratégias utilizadas por outros pesquisadores para balizar as decisões tomadas no presente trabalho.

1.2.2 Pesquisa Experimental

Segundo Chollet [5], todo trabalho de aprendizado de máquina segue um determinado fluxo de trabalho (*workflow*), que consiste em quatro passos: definição do problema, validação, *feature engineering* e redução de *overfitting*. Neste trabalho, será seguido o roteiro proposto por tal autor.

1.2.2.1 Definição do Problema

O objetivo deste trabalho é projetar um modelo de aprendizado de máquina capaz de prever a geração de um sistema fotovoltaico. A busca deste objetivo passa por três etapas: 1) a determinação das variáveis que influenciam a geração de energia de sistemas fotovoltaicos; 2) a obtenção de um conjunto de dados com estas variáveis e pré-processamento para uso em um modelo de aprendizado de máquina; 3) determinação do problema de aprendizado de máquina em si (neste caso, se trata de um problema de regressão) e escolha da arquitetura e parâmetros mais adequados.

2 FUNDAMENTACAO TEÓRICA

Este Capítulo trata da fundamentação teórica do presente trabalho. Nas Seções 2.1 e 2.2 são abordados brevemente o recurso solar e a geração fotovoltaica. Já na Seção 2.3 é feita uma apresentação de alguns dos principais conceitos de aprendizado de máquina.

2.1 ENERGIA SOLAR

2.1.1 O Recurso Solar

O Sol é a principal fonte de energia da Terra. Outras fontes derivam indiretamente da solar: por exemplo, o Sol causa a circulação atmosférica e o ciclo das águas, movimentando o vento e as águas aproveitados nas usinas eólicas e hidrelétricas. Combustíveis fósseis, bem como a matéria orgânica usada em processos de compostagem, são feitos a partir resíduos vegetais e animais, cuja cadeia alimentar é energizada primariamente pelo Sol. Pode-se dizer que estas formas de energia são solares indiretas e a energia utilizada em usinas solares e geradores fotovoltaicos é solar direta [6].

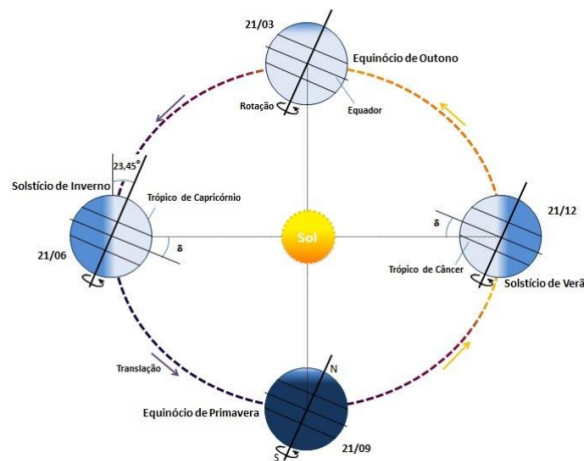
O aproveitamento da energia solar direta pode ser categorizado em duas formas principais: a forma passiva – como na arquitetura bioclimática, que aproveita a iluminação natural para economizar energia – ou a forma ativa, como no aquecimento de água em residências por meio de boilers e coletores solares. Outra forma ativa de aproveitamento da energia solar é na geração de energia elétrica: em usinas termosolares, concentradores solares aquecem fluidos para mover turbinas de geradores elétricos. Já na geração fotovoltaica, o que interessa é a luz em si, e não o calor, uma vez que, nas células fotovoltaicas, luz é convertida diretamente em eletricidade por meio do efeito fotovoltaico [6].

Radiação solar é um termo genérico para energia solar. Os termos mais específicos são: irradiância solar, para fluxo de potência (medido em W/m^2), e irradiação solar, para energia por unidade de área (medida em Wh/m^2). São relevantes para o estudo da radiação solar: a trajetória da Terra em torno do Sol; os ângulos que os raios solares formam com a superfície terrestre; a localização geográfica; as estações do ano e a duração dos dias [6].

A trajetória da Terra em torno do Sol é elíptica, com uma pequena

excentricidade. O eixo da Terra apresenta uma inclinação de aproximadamente $23,45^\circ$ em relação ao plano normal à elipse, o que faz com que o ângulo entre os raios solares e o plano do Equador variem entre $-23,45^\circ$ no solstício de inverno e $+23,45^\circ$ no solstício de verão (no hemisfério sul), passando por 0° nos equinócios de primavera e outono. Este ângulo, chamado Declinação Solar, é responsável pelas estações do ano e a variação da duração dos dias: os dias são mais curtos que a noite no inverno e mais longos no verão nos hemisférios Norte e Sul, já no outono e primavera e nas áreas próximas ao Equador eles têm aproximadamente mesma duração, conforme mostra a Figura 1 [6].

Figura 1: Órbita da Terra em torno do Sol



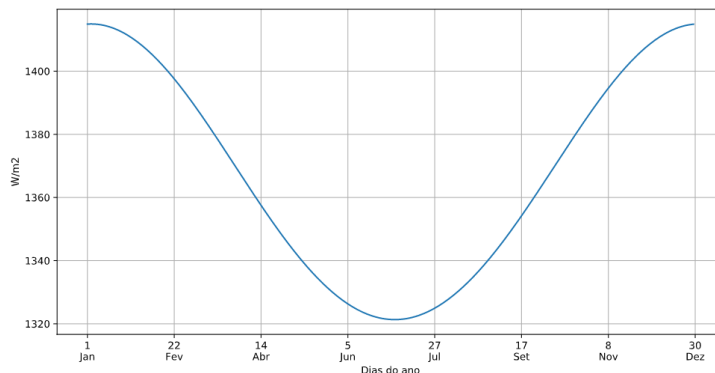
Fonte: PINHO e GALDINO, 2014, p. 70.

Em função da trajetória da Terra ao redor do Sol, a irradiância solar que incide no topo da atmosfera, chamada irradiância extraterrestre, tem um valor efetivo $I_{0,ef}$ que varia ao longo do ano, segundo a seguinte fórmula:

$$I_{0,ef} = I_0 \left(1 + 0,033 \cos \left(\frac{2\pi n}{365} \right) \right) \quad (1)$$

onde I_0 é a constante solar e n o dia do calendário juliano.

A constante solar I_0 é definida como o valor da irradiância extraterrestre anual média que incide sobre uma superfície perpendicular aos raios solares na distância média Terra-Sol, sendo aproximadamente 1367 W/m^2 [6]. A Figura 2 ilustra a variação da irradiância extraterrestre efetiva ao longo do ano no Hemisfério Sul.

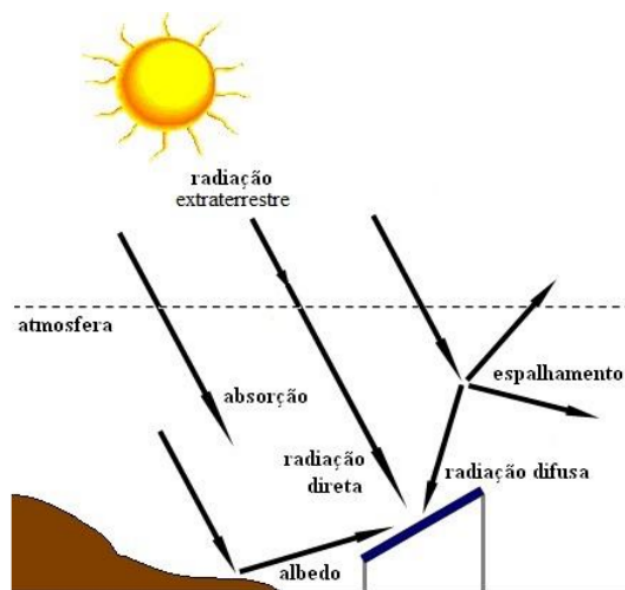
Figura 2: Variação da irradiância extraterrestre efetiva ao longo do ano

Fonte: Elaborada pelo autor.

A irradiância extraterrestre efetiva varia de 1320 a 1440 W/m^2 , com o valor mínimo ocorrendo no Solstício de Inverno e o máximo no Solstício de Verão [6]. 1367 W/m^2 , portanto, é o valor médio da irradiação solar que atinge o topo da atmosfera da Terra. Contudo, nem toda esta irradiação chega na superfície terrestre: a perda de fluxo de potência entre o topo da atmosfera e a superfície terrestre é de aproximadamente 27%, resultando em um valor médio de 1000 W/m^2 na superfície, razão pela qual este é o valor padrão para a especificação da potência nominal de células ou módulos fotovoltaicos [6].

A radiação solar que efetivamente chega na superfície terrestre é composta de três componentes: uma componente direta (ou de feixe), uma componente difusa e uma componente por reflexão, chamada de albedo, como ilustrado na Figura 3.

A componente direta vem da direção do Sol e a difusa é proveniente de todas as direções devido ao fenômeno de espalhamento da luz na atmosfera. Num dia de sol, pelo menos 20% da radiação que atinge uma superfície é difusa, enquanto que num dia nublado quase não há componente direta e a radiação difusa pode chegar a 100%. A terceira componente é devida à reflexão da luz nas superfícies do entorno. Como o coeficiente de reflexão de várias superfícies é chamado de albedo, esta componente também é chamada de componente devida ao albedo. A soma das três componentes é chamada de radiação global [6].

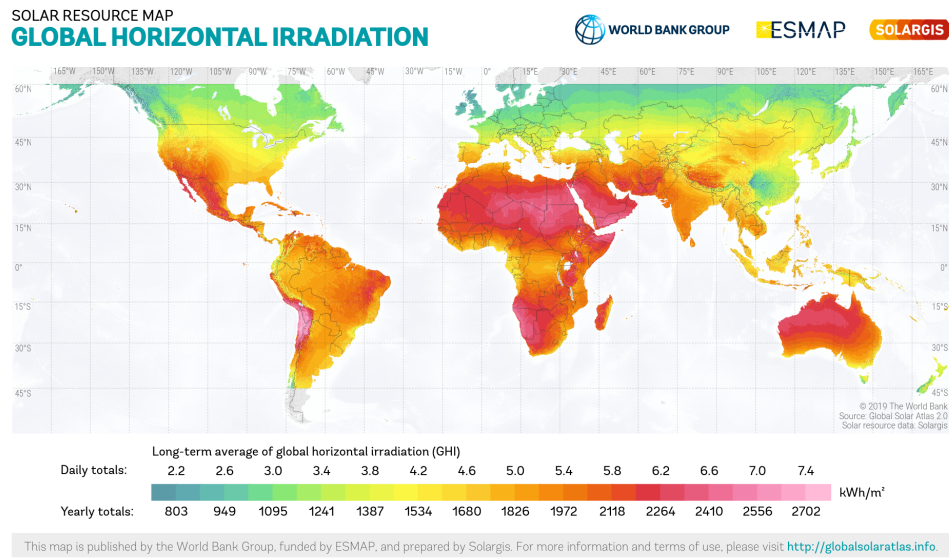
Figura 3: Componentes da radiação solar

Fonte: PINHO e GALDINO, 2014, p. 78

A distribuição da irradiância solar sobre a superfície terrestre é desigual, sendo as regiões mais próximas do Equador mais privilegiadas e as mais distantes, menos. A Figura 4 mostra distribuição da irradiação (Wh/m^2) normal (ou horizontal) global no mundo e a Figura 5 a irradiação normal no Brasil [7]. Florianópolis se encontra em uma das regiões menos ensolaradas do país, recebendo em média $4,2 kWh/m^2$ de irradiação solar global por dia, enquanto que as regiões Nordeste e Centro-Oeste são as mais ensolaradas [8].

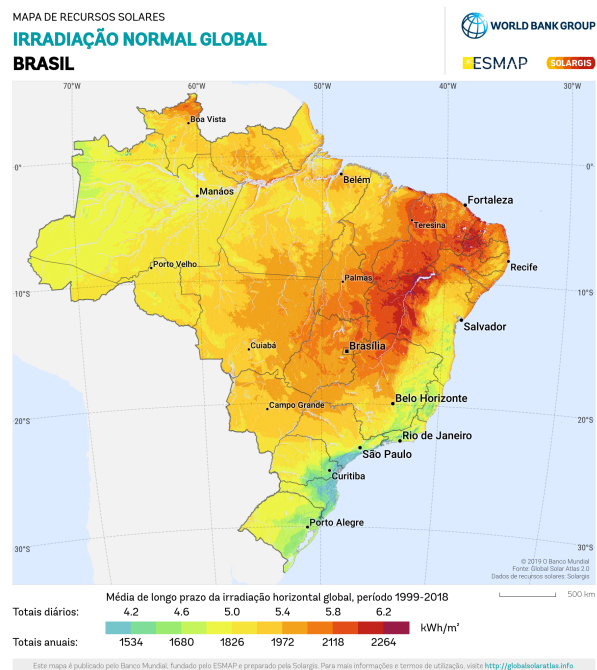
Pode-se medir a energia em horas de sol a pino (*full* ou *peak sun hours*) ao invés de Wh [9]. Quando o sol está a pino, a irradiância que atinge a superfície terrestre é de $1000 W/m^2$. Dividindo-se a irradiação total (em Wh/m^2) por $1000 W/m^2$, obtém-se o número de horas de sol a pino. Por exemplo, no caso de Florianópolis, considerando-se a irradiação diária de $4,2 kWh/m^2$ da Figura 5 e dividindo-se por $1000 W/m^2$ obtém-se $\frac{4,2kWh/m^2}{1kW/m^2} = 4,2h$. Esta medida expressa que, ao longo de todo o dia, a irradiação acumulada é equivalente a 4,2 horas de sol a pino.

Figura 4: Irradiação Normal Global – Mundo



Fonte: Global Solar Atlas [8]

Figura 5: Irradiação Normal Global – Brasil



Fonte: Global Solar Atlas [8]

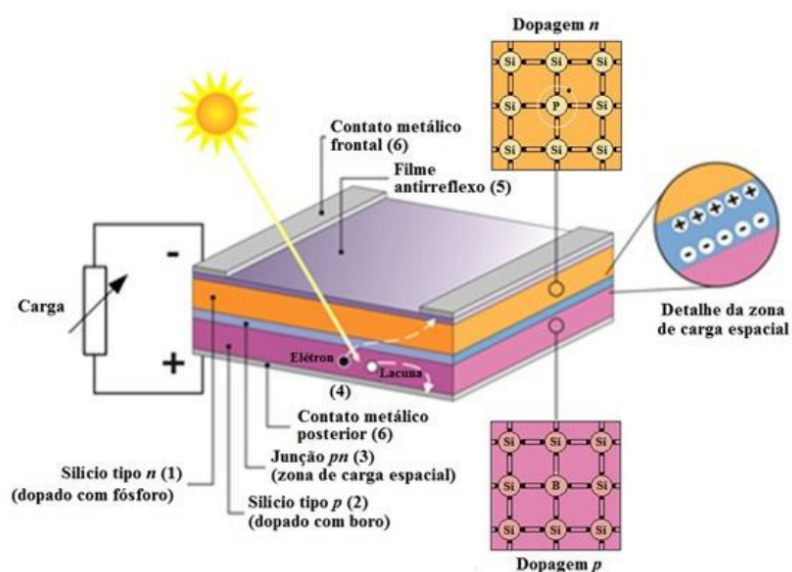
2.2 SISTEMAS FOTOVOLTAICOS

Sistemas fotovoltaicos convertem energia solar em energia elétrica por meio do efeito fotoelétrico. Um sistema fotovoltaico é constituído por um bloco gerador, um bloco de condicionamento de potência e um bloco de armazenamento (opcional). O bloco gerador é composto por módulos fotovoltaicos, cabeamento elétrico e estrutura de suporte. O bloco de condicionamento de potência pode ter conversores CC-CC, seguidor de ponto de potência máxima (SPPM), inversores, controladores de carga (se houver armazenamento) e outros dispositivos de proteção, supervisão e controle. Finalmente, o bloco de armazenamento é constituído por baterias e/ou outras formas de armazenamento [6]. Destes, apenas o bloco gerador é relevante para este trabalho. Mais especificamente: apenas o módulo e seu componente básico, a célula. Nesta seção, serão explanados o funcionamento da célula e do módulo fotovoltaicos.

2.2.1 A Célula Fotovoltaica

A célula fotovoltaica é o componente mais básico de um sistema fotovoltaico. Em sua construção mais comum, a célula consiste em duas lâminas de material semicondutor – geralmente silício (Si) – envolvidas por uma filme antirreflexo com contatos metálicos colocados de cada lado, como ilustrado na Figura 6.

Figura 6: Estrutura básica de uma célula fotovoltaica de silício



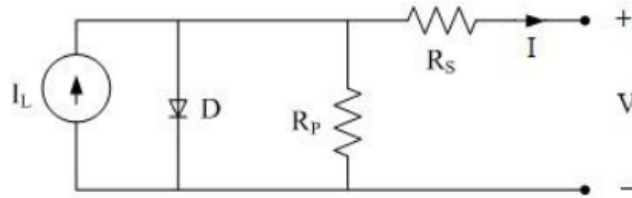
Fonte: PINHO e GALDINO, 2014, p. 112.

As lâminas de silício são especialmente dopadas para possuírem excesso de portadores de carga positiva e negativa (p e n): lacunas e elétrons, respectivamente. Uma zona de depleção é formada na interface entre as lâminas (junção pn), criando uma barreira de potencial entre as lâminas. Quando fótons com energia suficiente atingem a célula, pares elétron-lacuna são fotogerados e coletados pela junção pn: lacunas no lado p e elétrons no lado n, criando uma diferença de potencial entre os terminais da célula. Ao se prover uma via de condução entre os terminais, conectando-se os contatos metálicos de ambos os lados da célula, uma corrente contínua é formada.

Os fatores que limitam a eficiência de conversão de uma célula fotovoltaica são: 1) reflexão na superfície frontal; 2) sombra proporcionada pela área da malha metálica na face frontal; 3) absorção nula de fótons de energia menor que do que a banda de condução; 4) baixa probabilidade de coleta, pela junção pn, dos portadores de carga gerados fora da zona de depleção; 5) recombinação dos portadores de carga e 6) resistência elétrica no dispositivo e nos contatos metal-semicondutor, bem como possíveis caminhos de fuga da corrente (resistência em paralelo) [6].

As eficiências mais altas já obtidas em células fotovoltaicas estão na casa de 30%, com as melhores células de Si ficando em torno em de 25% em condições de laboratório [6].

No escuro, células fotovoltaicas se comportam exatamente igual a um diodo, razão pela qual o circuito equivalente pode ser modelado com um diodo. Além disso, como mencionado acima, a célula fotovoltaica possui resistências em série e em paralelo devido às resistências no dispositivo e dos contatos metálicos e fugas de corrente, respectivamente [6]. Sendo assim, a célula fotovoltaica pode ser representada pelo circuito equivalente mostrado na Figura 7. Em tal figura, D representa a junção pn no escuro, R_p e R_s as resistências em paralelo e série, I_L a corrente fotogerada e V e I , a tensão e corrente no terminal da célula.

Figura 7: Circuito equivalente básico para uma célula fotovoltaica

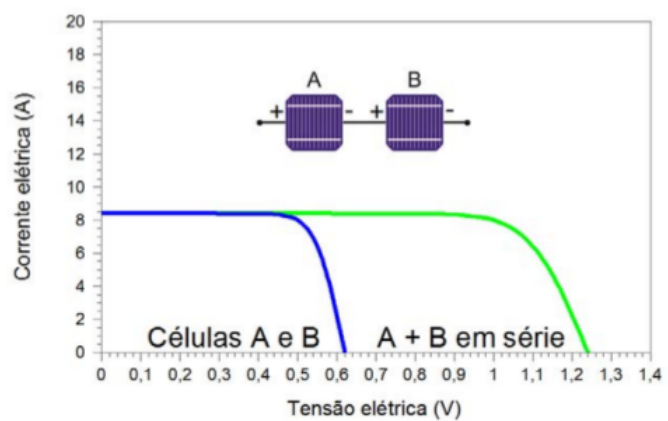
Fonte: PINHO e GALDINO, 2014, p. 118.

Os parâmetros elétricos da célula fotovoltaica são [6]:

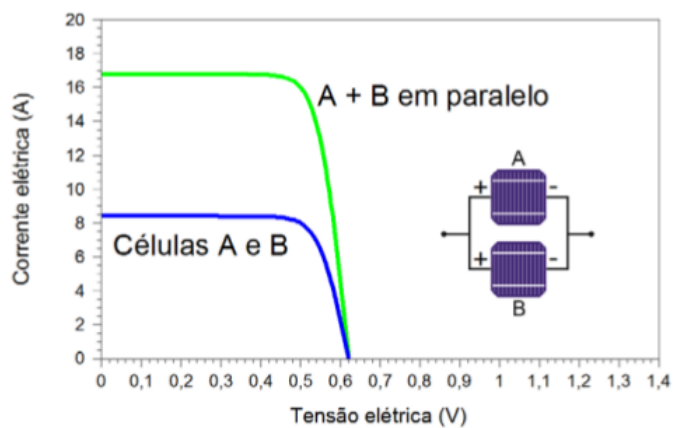
- **Tensão de circuito aberto (V_{oc}):** é a tensão entre os terminais de uma célula fotovoltaica quando não há corrente elétrica, e a tensão máxima da célula;
- **Corrente de circuito aberto (I_{sc}):** é a corrente da célula fotovoltaica quando a tensão elétrica em seus terminais é nula, e a corrente máxima da célula;
- **Fator de forma (FF):** é a razão entre a máxima potência da célula e o produto da corrente de curto circuito com a tensão de circuito aberto
- **Eficiência (η):** é a relação entre a potência elétrica produzida pela célula fotovoltaica e a potência da energia solar incidente.

Células podem ser conectadas em série e paralelo, como fontes de tensão. Na associação em série, somam-se as tensões e a corrente é constante e na associação em paralelo, a tensão é constante e somam-se as correntes, como mostra a Figura 8.

Figura 8: Curvas I-V de duas células fotovoltaicas ligadas em a) série e b) paralelo



(a)



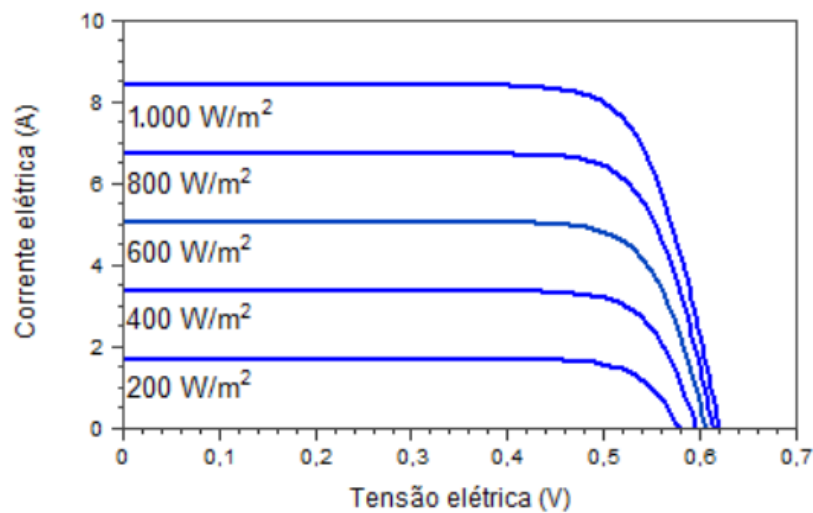
(b)

Fonte: PINHO e GALDINO, 2014, p. 122.

Entre os fatores externos que afetam o desempenho da célula, destacam-se a irradiância incidente e a temperatura de operação da mesma. Em caso de alta irradiância, a resistência em série R_s pode diminuir o desempenho se a célula não for projetada para estas condições, e em caso de baixa, a resistência em paralelo R_p pode diminuir o desempenho.

Em condições laboratoriais, a uma temperatura constante, a corrente aumenta linearmente com a irradiância e tensão de forma logarítmica. A Figura 9 mostra curvas I-V para diferentes valores de irradiância de uma célula fotovoltaica de silício cristalino a uma temperatura de 25 °C.

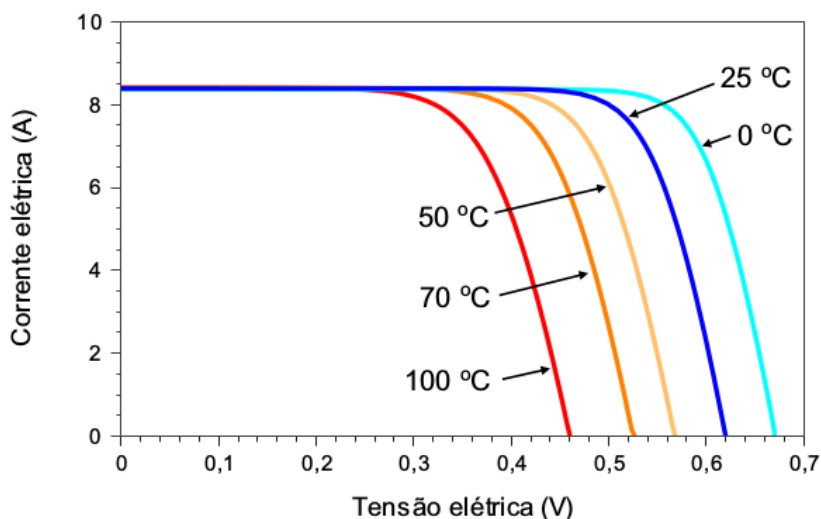
Figura 9: Influência da variação da irradiância solar na curva característica I-V de uma célula fotovoltaica de silício cristalino na temperatura de 25 °C



Fonte: PINHO e GALDINO, 2014, p. 126.

A própria irradiância e a temperatura ambiente afetam a temperatura de operação da célula. Com o aumento da temperatura, a tensão da célula reduz significativamente, porém a corrente aumenta muito pouco, resultando numa menor potência. A figura 10 mostra o mesmo para diferentes temperaturas e irradiância constante de 1000 W/m².

Figura 10: Influência da temperatura da célula fotovoltaica na curva I-V (para irradiância de 1000 W/m^2)



Fonte: PINHO e GALDINO, 2014, p. 127.

2.2.2 O Módulo Fotovoltaico

Módulos ou painéis fotovoltaicos são associações de células para a produção de energia elétrica e também proporcionam proteção mecânica e contra intempéries às células. São a forma como as células chegam ao consumidor final.

Células de silício podem ter valores de tensão de 0,3 a 0,8 V. Estas podem ser associadas em série ou paralelo em módulos para se obter diferentes valores de tensão e corrente. Módulos com tensão de 12 V são utilizados para carregar baterias, e podem ser associados em série para sistemas de 24 e 48 V em corrente contínua. Para outras aplicações, valores entre 30 a 120 V são mais comuns [6]. Os módulos também podem ser associados em série e paralelo, exatamente igual às células.

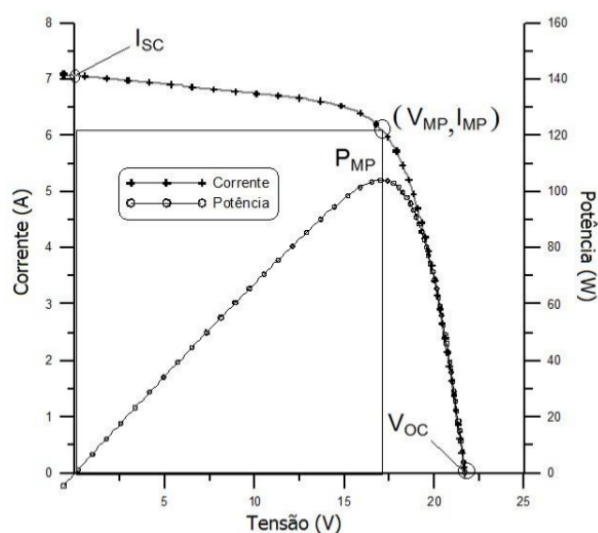
Módulos possuem alguns dos mesmos parâmetros elétricos que células: V_{oc} , I_{sc} , fator de forma (FF) e eficiência η . Além destes, o módulo também é caracterizado pela potência elétrica de pico (W_p), isto é a potência gerada pelo módulo em condições-padrão de ensaio (STC, do inglês *Standard Test Conditions*): irradiância de 1000 W/m^2 e temperatura de 25 °C.

Pode-se obter a curva característica I-V de um módulo submetendo-o, em condições-padrão de ensaio, a uma fonte de tensão variável. Com este ensaio, obtém-se ainda os parâmetros:

- **Ponto de máxima potência (P_{MP}):** ponto na curva de potência correspondente ao produto de V_{MP} e I_{MP} e potência máxima gerada pelo módulo;
- **Tensão de máxima potência (V_{MP}):** tensão nos terminais do módulo quando este está na potência máxima;
- **Corrente de máxima potência (I_{MP}):** corrente quando o módulo está na potência máxima.

A curva característica I-V, bem como a curva de potência P-V e os parâmetros de um módulo fotovoltaico são ilustrados pela Figura 11.

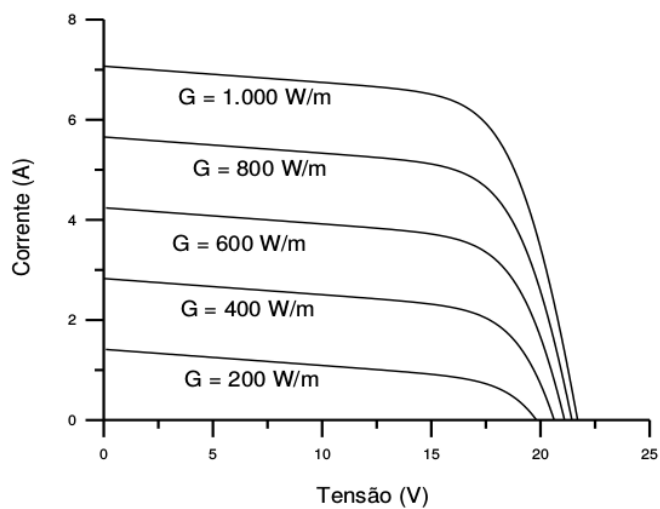
Figura 11: Curva característica I-V e curva de potência P-V para um módulo com potência nominal de 100 Wp



Fonte: PINHO e GALDINO, 2014, p. 147.

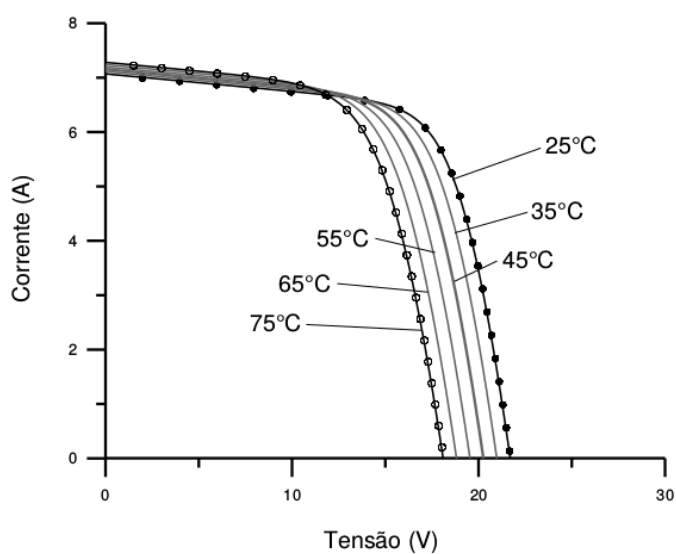
Sendo composto basicamente de células, o módulo é semelhantemente afetado pela irradiância solar e temperatura de operação. As Figuras 11 e 12 mostram as curvas I-V de um módulo fotovoltaico submetido a variações de irradiância e temperatura, respectivamente.

Figura 12: Efeito da variação da irradiância solar sobre a curva característica I-V para um módulo fotovoltaico de 36 células de silício cristalino (c-Si) a 25 °C



Fonte: PINHO e GALDINO, 2014, p. 149.

Figura 13: Efeito da variação da temperatura das células sobre a curva característica I-V para um módulo fotovoltaico de 36 células de silício cristalino (c-Si) sob irradiância de 1000 W/m²



Fonte: PINHO e GALDINO, 2014, p. 150.

2.3 APRENDIZADO DE MÁQUINA

2.3.1 Visão Geral de Inteligência Artificial, Aprendizado de Máquina e Aprendizado Profundo

No contexto do presente trabalho, é interessante o entendimento dos conceitos de inteligência artificial, aprendizado de máquina e aprendizado profundo, conforme discutido em Chollet [5]. O Diagrama de Venn na Figura 14 relaciona esses conceitos. Note-se que um é embarcado no outro: aprendizado profundo é um subtipo de aprendizado de máquina, e ambos são subtipos de inteligência artificial.

O conceito de inteligência artificial (IA) é o mais abrangente dos três. A inteligência artificial surgiu nos anos 1950 no campo da ciência da computação e consiste em tentar automatizar tarefas intelectuais normalmente realizadas por humanos. Ela abarca aprendizado de máquina e aprendizado profundo, mas também inclui abordagens que não envolvem qualquer tipo de aprendizado.

Figura 14: Relação entre IA, aprendizado de máquina e aprendizado profundo



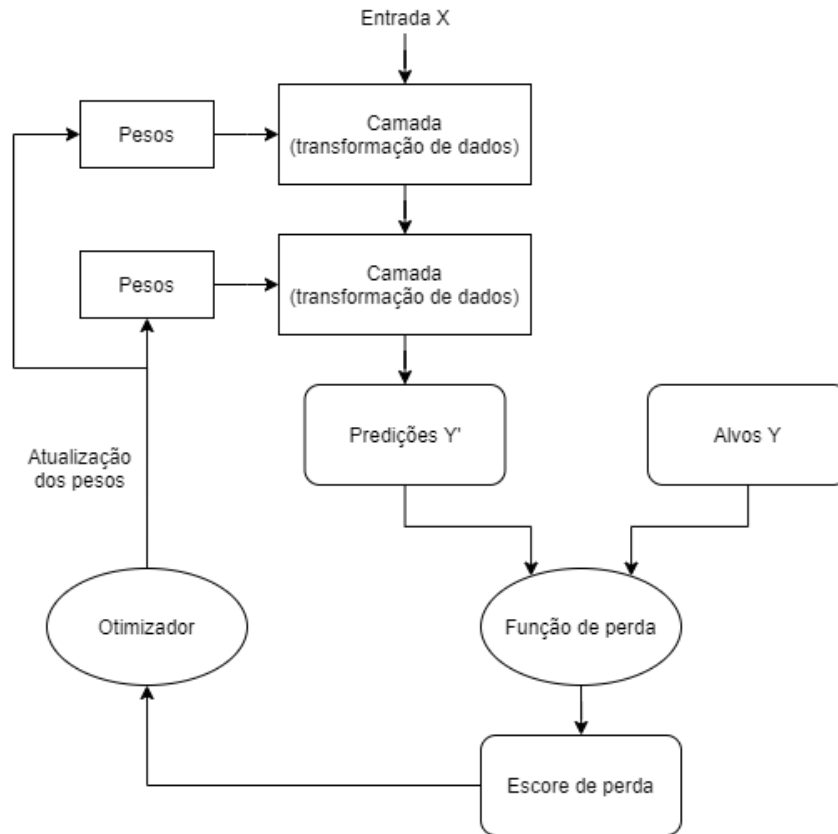
Fonte: Elaborada pelo autor, adaptada de CHOLLET, 2018, p. 4.

No aprendizado de máquina, ao invés de o computador processar dados segundo um método pré-definido – usando codificação rígida (*hard coding*) –, mostra-se ao computador uma série de exemplos de entradas e saídas e ele deduz o método a partir dos dados. Espera-se que um modelo de aprendizado de máquina receba dados de entrada e retorne saídas que façam sentido. Chollet [5] afirma que o problema central do aprendizado de máquina é a transformação significativa de dados: em outras palavras, como se obter representações significativas dos dados de entrada de modo a facilitar a obtenção da saída esperada. Isto é feito submetendo-se os dados a uma ou mais de uma série de operações

matemáticas, como mudanças de coordenadas, transformações lineares, não lineares, etc. A performance do algoritmo – calculada por meio de alguma métrica, como acurácia, erro quadrático médio, etc. – é então usada como sinal de feedback a fim de se obter a melhor representação. O “aprendizado” na expressão “aprendizado de máquina”, sumariza o autor, consiste na automatização deste processo.

O aprendizado profundo é uma abordagem de aprendizado de máquina que utiliza sucessivas camadas de representações significativas de dados [5]. Estas camadas, explica o autor, são “empilhadas” umas sobre as outras e treinadas utilizando-se um modelo chamado de rede neural. Em cada camada da rede neural os dados são submetidos a transformações parametrizadas por pesos (também chamados de parâmetros de camada). A saída da última camada é uma previsão e esta é comparada a um alvo – o valor esperado – por meio de uma função de perda (ou função objetiva). Esta função retorna uma pontuação – chamada de score de perda ou de distância – que expressa a diferença entre o valor predito e o valor esperado. Liga-se então a saída da função de perda a um otimizador, que realiza um processo chamado de propagação reversa (*backpropagation*), que atualiza os valores dos pesos. O treinamento da rede neural é realizado em ciclos: os pesos são inicializados com valores aleatórios e a cada rodada de treinamento eles são ajustados de modo a diminuir progressivamente a perda. O objetivo do treinamento da rede neural é obter valores de pesos que minimizem a função de perda.

O diagrama na Figura 15 ilustra o funcionamento básico da rede neural.

Figura 15: Diagrama de blocos de uma rede neural

Fonte: Elaborada pelo autor, adaptada de CHOLLET, 2018, p. 11.

2.3.2 Aprendizagem de Máquina

A noção do aprendizado de máquina consiste em ensinar uma máquina a realizar uma tarefa computacional observando pontos de dados. É um campo relativamente recente e suas aplicações vão desde regressão, detecção de objetos em imagens e reconhecimento de fala até diagnóstico de doenças e previsões de preços em mercados de ações, com mais aplicações sendo descobertas a todo momento [10].

O tipo mais comum de aprendizado de máquina é o chamado aprendizado supervisionado. Este envolve um conjunto de pares entrada-saída $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N$ chamado de conjunto de treinamento, onde N é o número de exemplos (ou pontos de dados), e o objetivo do aprendizado é mapear as entradas \mathbf{x} às saídas y [11]. Há ainda aprendizado não supervisionado e aprendizado por reforço, porém estes métodos não serão tratados neste trabalho. Finalmente, modelos de aprendizado de máquina

podem ser paramétricos ou não paramétricos: isto é, possuem um número fixo de atributos ou o número de atributos aumenta conforme aumentam os dados de treinamento, respectivamente [11].

Pode-se formalizar o aprendizado como uma aproximação de funções. Assume-se que para cada par (\mathbf{x}, y) , existe uma função f desconhecida tal que $y = f(\mathbf{x})$. Trata-se então de descobrir uma função \hat{f} que se aproxime de f , de modo que as previsões do modelo sejam dadas por $\hat{y} = \hat{f}(\mathbf{x})$.

O vetor de entrada (ou *input*) \mathbf{x} pode ser, no caso mais simples, um vetor D -dimensional de números representando atributos (também chamados de *features*, características ou covariáveis). Frequentemente, porém, ele é uma estrutura complexa – *e.g.* uma imagem colorida digitalizada representada por um tensor de três matrizes – e cabe ao projetista deixar esta estrutura de uma forma utilizável pelo algoritmo.

A variável de saída y (também chamada de *output*, rótulo, alvo ou resposta) é geralmente uma variável categorial ou nominal de um conjunto finito ou um número escalar real. No primeiro caso o problema se trata de uma classificação e no segundo, de uma regressão.

Há, é claro, uma variedade de tarefas de aprendizado de máquina além de classificação e regressão [12]. Porém, uma exposição exaustiva de todas elas fugiria ao escopo deste trabalho.

A tarefa da classificação consiste em relacionar \mathbf{x} a $y \mid y \in \{1, \dots, C\}$, onde C é o número de classes. Se $C = 2$, trata-se de uma classificação binária; se $C > 2$ e as classes forem mutuamente exclusivas, então a classificação é chamada de multiclasse ou multinomial; e se $C > 2$ e as classes não forem mutuamente exclusivas, chama-se classificação multi-rótulo (*multilabel*).

A regressão, por sua vez, é quase idêntica à classificação, exceto que a variável de saída é contínua ao invés de discreta [11]. Neste trabalho, são discutidos dois tipos de regressão: linear e logística.

2.3.2.1 Regressão Linear

Na regressão linear, assume-se que a saída y é uma função linear da entrada \mathbf{x} :

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} + b = \sum_{j=1}^m w_j x_j + b \quad (2)$$

onde $\mathbf{x} = [x_1, x_2, \dots, x_m]^T$ é o vetor de entrada com m atributos; $\mathbf{w} = [w_1, w_2, \dots, w_m]^T$ é o vetor de parâmetros (ou coeficientes ou pesos); b é uma constante e $\mathbf{w}^T \mathbf{x}$ é um produto escalar.

Fazendo $b = w_0$ é possível reescrever a equação como:

$$y(\mathbf{x}) = \mathbf{w}^T \mathbf{x} \quad (3)$$

onde $\mathbf{x} = [1, x_1, \dots, x_m]^T$ e $\mathbf{w} = [w_0, w_1, \dots, w_m]^T$, ambos vetores de dimensão $M = m + 1$.

Se \hat{y} for a previsão do modelo, define-se o erro residual como $\epsilon = y - \hat{y}$ [11].

A Equação (2) representa um único exemplo ou ponto de dado (\mathbf{x}, y) . Para representar todos os exemplos do conjunto, porém, é mais conveniente utilizar a notação vetorial:

$$\mathbf{y} = \mathbf{X}\mathbf{w} \quad (4)$$

onde $\mathbf{X} = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_N^T \end{bmatrix}$ é uma matriz $N \times M$ contendo N vetores de entrada; $\mathbf{w} = [w_0, w_1, \dots, w_m]^T$ é um vetor $M \times 1$; e $\mathbf{y} = [y_1, \dots, y_N]^T$ é um vetor $N \times 1$ contendo N saídas. Cada linha da matriz \mathbf{X} é um vetor coluna $M \times 1$ transposto em um vetor linha $1 \times M$.

Deseja-se encontrar os parâmetros \mathbf{w} que minimizem o módulo do erro residual $\epsilon = y - \hat{y}$. Uma maneira de se fazer isso é, primeiramente, definindo-se uma medida deste erro em função de \mathbf{w} . Convencionalmente, esta função é chamada de função de perda (*loss function*) $L(y, \hat{y})$ quando aplicada a um único exemplo e função de custo (*cost function*) $J(\mathbf{y}, \hat{\mathbf{y}})$ quando aplicada a todos os elementos do conjunto.

Para o modelo linear, uma função de custo adequada – assumindo-se uma distribuição normal dos dados e deduzida utilizando-se o método de máxima verossimilhança (*maximum-likelihood estimation* – MLE) [11] – é soma dos quadrados dos resíduos SQR (*residual sum of squares* – RSS):

$$J(\mathbf{y}, \hat{\mathbf{y}}) = J(\mathbf{y}, \mathbf{w}^T \mathbf{x}) = J(\mathbf{w}) = \sum_{i=1}^N (y_i - \mathbf{w}^T \mathbf{x}_i)^2. \quad (5)$$

Se for tirada a média da SQR, esta métrica se torna o erro quadrático médio EQM (*mean squared error* – MSE). A SQR também pode ser escrita como o quadrado

da norma L2 do vetor de erros residuais ϵ :

$$J(\mathbf{w}) = \|\epsilon\|_2^2 = \sum_{i=1}^N \epsilon_i^2 \quad (6)$$

onde $\epsilon_i = y_i - \mathbf{w}^T \mathbf{x}_i$.

Como o objetivo é minimizar a soma dos quadrados dos resíduos, este método é chamado de mínimos quadrados (*least squares*).

Em seguida, calcula-se o gradiente da função custo para se descobrir o ponto de mínimo global da função. Reescrevendo-se $J(\mathbf{w})$ em notação vetorial, tem-se

$$J(\mathbf{w}) = \frac{1}{2}(\mathbf{y} - \mathbf{X}\mathbf{w})^T(\mathbf{y} - \mathbf{X}\mathbf{w}) = \frac{1}{2}\mathbf{w}^T(\mathbf{X}^T\mathbf{X})\mathbf{w} - \mathbf{w}^T(\mathbf{X}^T\mathbf{y}). \quad (7)$$

Calculando-se o gradiente de (6) em relação a \mathbf{w} , obtém-se

$$g(\mathbf{w}) = \mathbf{X}^T\mathbf{X}\mathbf{w} - \mathbf{X}^T\mathbf{y} = \sum_{i=1}^N \mathbf{x}_i(\mathbf{w}^T \mathbf{x}_i - y_i). \quad (8)$$

O ponto onde o gradiente é nulo é o ponto de mínimo. Igualando-se (7) a zero, tem-se

$$\mathbf{X}^T\mathbf{X}\mathbf{w} = \mathbf{X}^T\mathbf{y}. \quad (9)$$

A solução $\hat{\mathbf{w}}$ deste sistema é dada por

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y} \quad (10)$$

em que $\hat{\mathbf{w}}$ é o que se procura: o vetor de parâmetros que minimiza o erro residual.

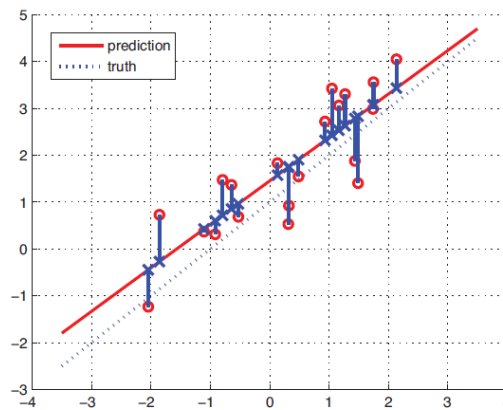
O caso mais simples de regressão linear é o unidimensional, onde y é uma função de primeiro grau:

$$y(\mathbf{x}) = w_0 + w_1x = \mathbf{w}^T \mathbf{x}. \quad (11)$$

Neste exemplo, $\mathbf{x} = [1, x]^T$ é um vetor de entrada com um único atributo e $\mathbf{w} = [w_0, w_1]^T$ é o vetor de parâmetros, sendo $w_0 = b$ o coeficiente angular (ou *bias*) e w_1 o coeficiente linear da equação. A Figura 16 ilustra este exemplo: os pontos de dados (x_i, y_i) são representados por círculos vermelhos; os valores estimados (x_i, \hat{y}_i) por cruzes

azuis; e os erros residuais $\epsilon = (y_i - \hat{y}_i)$ por linhas azuis verticais. O objetivo é achar os parâmetros (w_0 e w_1) tais que a linha vermelha resultante minimize a soma de resíduos quadrados [11].

Figura 16: Distâncias entre os pontos de dados e as predições. O método dos mínimos quadrados visa minimizar a soma dos quadrados destas distâncias



Fonte: MURPHY, 2012, p. 219.

O modelo linear assume que os dados podem ser separados por uma linha (ou hiperplano). Isto nem sempre é verdade; contudo, ainda é possível separar dados não-linearmente separáveis substituindo-se \mathbf{x} por uma função de base não-linear $\phi(\mathbf{x})$ [13]. Este procedimento se chama expansão da função de base (*basis function expansion*). No caso em que $\phi(\mathbf{x}) = [1, x, x^2, \dots, x^d]$, trata-se de uma regressão polinomial, porém ϕ pode ser qualquer função não-linear de \mathbf{x} .

Murphy nota que muitas abordagens populares de aprendizado de máquina – como máquinas de vetores de suporte, redes neurais, árvores de regressão e classificação – podem ser vistas como métodos de se estimar funções de base a partir dos dados [11].

2.3.2.2 Regressão Logística

Pode-se usar a regressão linear para se realizar uma classificação binária – $y \in \{0, 1\}$ – utilizando a combinação linear $\mathbf{w}^T \mathbf{x}$ como parâmetro de uma função logística

(ou sigmoide) [11]:

$$y(\mathbf{x}) = \sigma(\mathbf{w}^T \mathbf{x}) \quad (12)$$

onde $\sigma(z) = \frac{1}{1+e^{-z}}$.

A função logística “achata” a curva da resposta, forçando-a a assumir um valor entre 0 e 1. Um esboço da função é dado na Figura 17. Nota-se que $\sigma(-\infty) = 0$, $\sigma(0) = 0.5$ e $\sigma(\infty) = 1$. Se o valor central da função, $\sigma(0) = 0.5$, for usado como limiar (*threshold*), pode-se realizar uma classificação binária com a seguinte regra de decisão:

$$y(\mathbf{x}) = \begin{cases} 1 & \sigma(\mathbf{w}^T \mathbf{x}) > 0.5; \\ 0 & \text{caso contrário.} \end{cases} \quad (13)$$

Com relação à escolha da função custo, não é possível utilizar a SQR com a função logística, pois resulta em uma função custo não-convexa. Convexidade significa que a curva (ou hiperplano) de uma função é “voltada para baixo” e ela possui um único ponto estacionário – onde a derivada ou gradiente são nulos – e ele coincide com o ponto de mínimo global da função. Mais formalmente, uma função $g(\mathbf{w})$ é dita convexa se for duas vezes diferenciável e sua segunda derivada $g''(\mathbf{w}) \geq 0$ para todo \mathbf{w} [10]. A relevância da convexidade da função custo em aprendizado de máquina é que se a função custo tiver pontos estacionários além do ponto de mínima global, o algoritmo pode parar sem ter realmente minimizado o erro residual. Por esta razão, utiliza-se como função perda a entropia cruzada $L(y, \hat{y}) = -\sum_{i=1}^N [y_i \log \hat{y}_i + (1 - y_i) \log(1 - \hat{y}_i)]$, resultando na seguinte função custo (em notação vetorial):

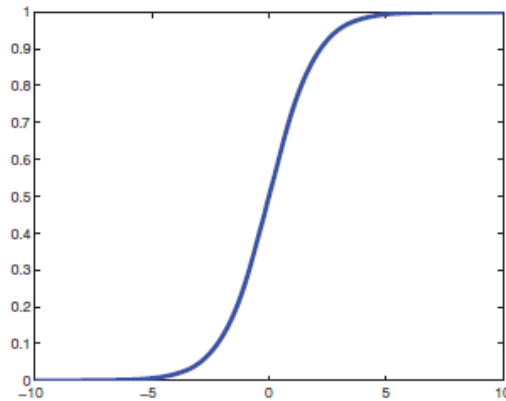
$$J(\mathbf{w}) = \mathbf{y}^T \log \hat{\mathbf{y}} - (\mathbf{1} - \mathbf{y})^T (\mathbf{1} - \hat{\mathbf{y}}) \quad (14)$$

cujo gradiente é

$$g(\mathbf{w}) = \mathbf{x}^T (\hat{\mathbf{y}} - \mathbf{y}) = \mathbf{x}^T (\sigma(\mathbf{x}\mathbf{w}) - \mathbf{y}). \quad (15)$$

Segundo Murphy [11], não é possível escrever esta expressão de forma fechada, sendo portanto necessário recorrer a algoritmos de otimização para se achar a solução.

Figura 17: Função Logística



Fonte: MURPHY, 2012, p. 21.

2.3.2.3 Otimização

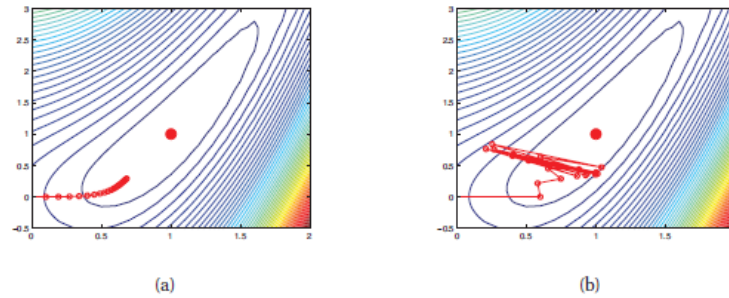
O objetivo da otimização é, dada uma função custo $J(\boldsymbol{\theta})$, descobrir os parâmetros $\boldsymbol{\theta}$ que minimizem J . Existe uma variedade de métodos de otimização. O método mais utilizado para aprendizado de máquina em geral e aprendizado profundo em particular é o método do gradiente descendente estocástico [12, 294], que pode ser descrito desta forma:

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k - \eta_k \mathbf{g}_k \quad (16)$$

onde $\boldsymbol{\theta}$ é o vetor de parâmetros (na Equação (14), $\boldsymbol{\theta} = \mathbf{w}$); \mathbf{g}_k é o gradiente da função custo; e η_k é a taxa de aprendizado (*learning rate* ou *step size*), isto é, a magnitude do ajuste dos parâmetros a cada ciclo do algoritmo.

A escolha da taxa de aprendizado é importante, uma vez que uma taxa muito pequena resultará numa convergência muito lenta, e uma taxa muito grande poderá resultar em divergência, conforme ilustra a Figura 18.

Figura 18: 20 ciclos do método do gradiente a taxas de aprendizado a) $\eta = 0.1$ e b) $\eta = 0.6$. O algoritmo começa no ponto $(0,0)$ e o ponto de mínimo global é $(1,1)$



Fonte: (MURPHY, 2012, p. 247).

Segundo Chollet [5], há um compromisso entre otimização e overfitting, sendo que quanto mais otimizado um modelo de aprendizado de máquina for para o conjunto de treinamento, maior tende a ser o overfitting e vice-versa. O objetivo do projetista é obter generalidade: equilíbrio entre otimização e overfitting.

2.3.2.4 Overfitting

Uma questão de fundamental importância em estatística e aprendizado de máquina diz respeito ao conceito de overfitting. Uma tradução de overfitting poderia ser “sobreajuste”. Diz-se que um modelo está “sobreajustado” a um conjunto quando ele representa bem os pontos de dados deste conjunto, mas não sua real distribuição fora dele. Segundo Mostafa [13], overfitting ocorre quando o modelo é mais complexo do que o necessário para representar a função alvo e acaba utilizando sua capacidade adicional para representar idiosincrasias do conjunto, como erros, ruído e valores aberrantes (*outliers*).

2.3.3 Aprendizado Profundo

Os modelos mais comuns de aprendizado profundo são as chamadas redes neurais feedforward (*feedforward neural networks*) ou perceptron multicamadas (*multilayer perceptrons*). O funcionamento desta família de modelos pode ser resumido da seguinte forma: dada uma função desconhecida $y = f(\mathbf{x})$, a rede define uma nova função $\hat{y} = g(\mathbf{x}, \boldsymbol{\theta})$ que mapeia os dados \mathbf{x} e certos parâmetros $\boldsymbol{\theta}$ ao alvo \hat{y} e, através do

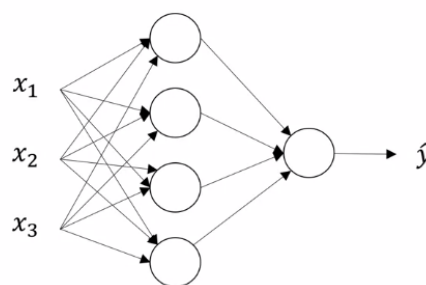
processo de treinamento, determina os parâmetros θ que resultam na melhor aproximação de \hat{y} à função original y . Estas redes são chamadas de feedforward porque não possuem retroalimentação. Quando redes neurais possuem retroalimentação, são chamados de redes neurais recorrentes (*recurrent neural networks* – RNN) [12, 168]. Existem ainda redes neurais convolucionais (*convolutional neural networks* – CNN), ideais para dados que possuem uma topologia em grade, como imagens digitais [12, 330]. Neste trabalho, tratar-se-á exclusivamente de redes neurais feedforward.

2.3.3.1 Rede Neural Feedforward

Na apresentação do funcionamento da rede neural feedforward, pode-se aproveitar bastante do que foi dito na parte sobre regressão logística. De fato, uma rede neural feedforward nada mais é do que “uma série de modelos de regressão logística empilhados uns sobre os outros, com a última camada sendo um outro modelo de regressão logística ou um modelo de regressão linear, dependendo de se estamos resolvendo um problema de classificação ou regressão” [11, 563].

Redes neurais feedforward são geralmente representadas por diagramas esquemáticos como o da Figura 19.

Figura 19: Rede neural de duas camadas



Fonte: Curso Neural Networks and Deep Learning [14].

A profundidade da rede diz respeito ao número de camadas, L . A rede

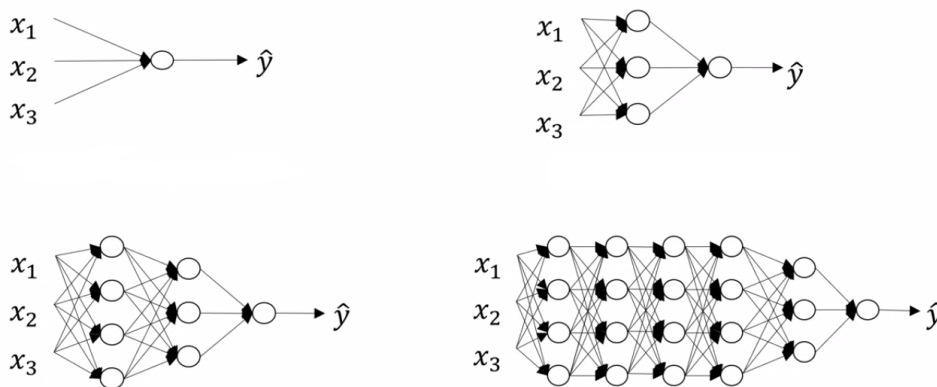
¹a series of logistic regression models stacked on top of each other, with the final layer being either another logistic regression or a linear regression model, depending on whether we are solving a classification or regression problem.

representada acima tem apenas duas camadas: a primeira com quatro unidades e a segunda com uma unidade. Uma rede neural com apenas uma camada é idêntica a uma regressão logística (para ser “profunda” ela precisa ter pelo menos duas). O verdadeiro potencial do aprendizado profundo se faz notar quando se aumenta a complexidade do modelo, acrescentando-se mais camadas e/ou mais unidades por camada (com a ressalva de que redes mais complexas do que o necessário podem causar overfitting) [5, 8].

O número de camadas é contado da camada imediatamente após a entrada até a última (a entrada – o vetor, matriz ou tensor contendo os atributos \mathbf{x} – não conta como camada). A saída, ou última camada, também é a saída da rede como um todo. Camadas entre a entrada e a saída são chamadas de camadas ocultas (*hidden layers*), sempre em número de $L - 1$.

A Figura 20 mostra exemplos de redes neurais com diferentes números de camadas.

Figura 20: Redes neurais com uma, duas (uma oculta), três (duas ocultas) e seis (cinco ocultas) camadas



Fonte: Curso Neural Networks and Deep Learning [14].

Neste trabalho é utilizada a seguinte convenção:

- L : número de camadas;
- l : índice de uma camada específica. Por exemplo, $l = 2$ refere-se à segunda camada;
- $n^{[l]}$: número de elementos em uma camada. Por exemplo, $n^{[2]} = 4$ significa que a camada 2 possui 4 elementos.

Nesta convenção, a entrada pode ser representada por $l = 0$ e seu número de elementos por $n^{[0]}$ ou n_x . Logo, $n^{[0]} = n_x = m$ equivale a dizer que a entrada do modelo possui m atributos.

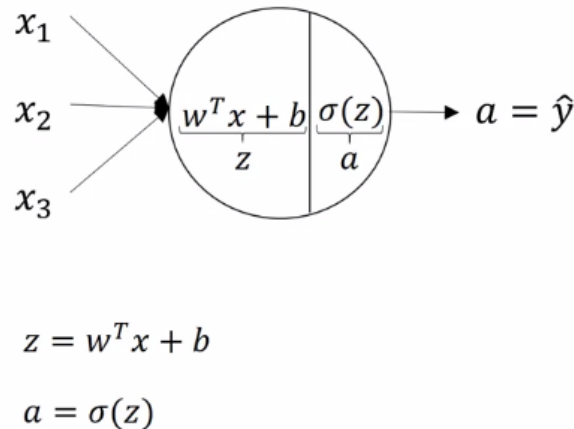
Cada camada $l = 1, \dots, L$ possui $n^{[l]}$ elementos, chamados de unidades de camada ou neurônios. Estes elementos são representados pelos círculos nos diagramas das Figuras 19 e 20.

A camada de saída possui $n^{[L]} = 1$ em se tratando de uma regressão, $n^{[L]} = 2$ em uma classificação binária e $n^{[L]} = K$ em uma classificação multinomial.

O número de elementos das camadas ocultas varia, ficando a cargo do projetista.

Com relação às unidades de camada ou neurônios, cada uma se subdivide em uma combinação linear e uma função de ativação, conforme ilustrado na Figura 21.

Figura 21: Unidade de Camada ou Neurônio



Fonte: Curso Neural Networks and Deep Learning [14].

Na Figura 21, a equação $z = \mathbf{w}^T \mathbf{x} + b$ refere-se à combinação linear e a equação $a = \sigma(z)$ à função de ativação. Nota-se que este neurônio representa um modelo de regressão logística, idêntico ao discutido na Seção 2.3.2.2.

Pode-se indexar cada neurônio da rede neural empregando-se os sobrescritos $[l]$ e (i) , onde $[l] = 0, \dots, L$ denota a camada e o $(i) = 1, \dots, n^{[l]}$ o neurônio. Deste modo, pelo

sobrescrito $[l](i) = 1, 2$ entende-se o segundo neurônio (de cima pra baixo) da primeira camada, e assim por diante.

Similarmente, cada equação da rede neural pode ser indexada da seguinte forma:

$$z^{[l](i)} = \mathbf{w}^{T[l](i)} \mathbf{x}^{[l](i)} + b^{[l](i)} \quad (17)$$

e

$$a^{[l](i)} = \sigma^{[l]}(z^{[l](i)}) = \sigma^{[l]}(\mathbf{w}^{T[l](i)} \mathbf{x}^{[l](i)} + b^{[l](i)}) \quad (18)$$

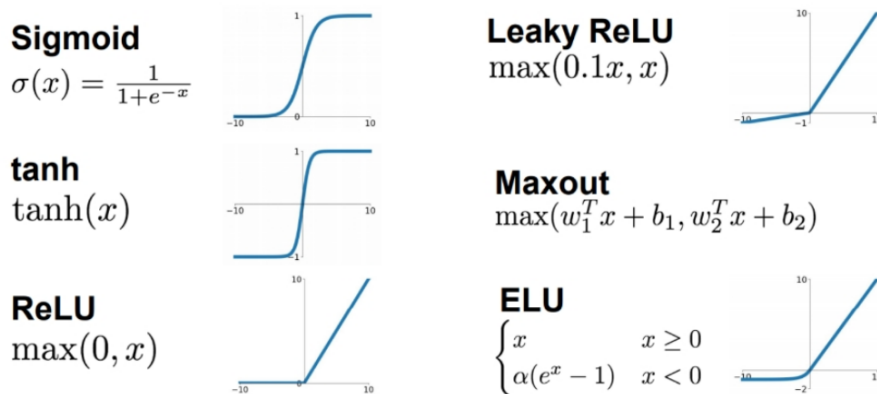
Em todas as camadas, exceto a última, a função de ativação é a função logística ou outra função não-linear. Na última camada, porém, ela pode ser uma função não-linear, em caso de classificação, ou a função identidade, em se tratando de uma regressão. Na ausência de funções de ativação não-lineares nas unidades de camada, a rede seria uma combinação de combinações lineares – portanto também uma combinação linear – e não apresentaria nenhuma vantagem adicional com relação aos modelos já discutidos anteriormente.

Além da função logística, outras funções de ativação bastante utilizadas são: tangente hiperbólica (\tanh) e retificadora ou *ReLU* (do inglês *Rectified Linear Unit*). O gráfico da função \tanh é similar ao gráfico da função logística (Figura 22), porém é centralizada no zero e satura de $[-1, 1]$, enquanto que a logística não é centralizada no zero e satura de $[0, 1]$.

Uma limitação que ambas as funções logística e \tanh apresentam é a saturação. Isto faz com que estas funções tenham gradientes próximos de zero para valores positivos ou negativos muito altos de z , resultando em uma performance lenta do algoritmo de otimização. A função *ReLU*, $g(z) = \max(0, z)$, contorna este problema, sendo linear para valores positivos de z e não apresentando saturação. De fato, *ReLU* é a função de ativação padrão mais recomendada para redes neurais atualmente [12, 174].

A Figura 22 apresenta o gráfico das funções ativação sigmoide, \tanh , *ReLU* e outras.

Figura 22: Funções de ativação



Fonte: Blog Technical Fridays [15].

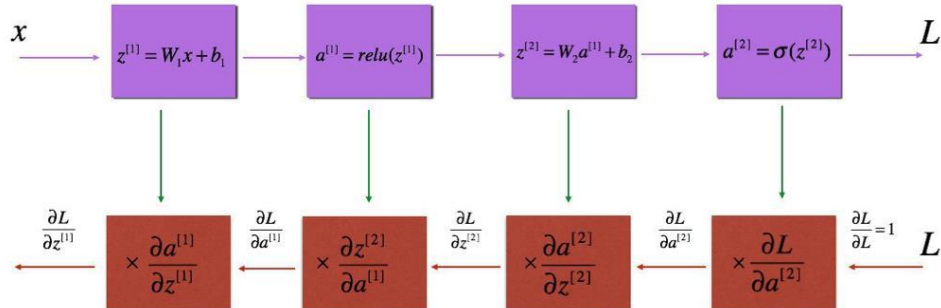
2.3.3.2 Outros Métodos de Otimização

Segundo Goodfellow et al. [12, 274], algoritmos de otimização para redes neurais profundas diferem de algoritmos de otimização utilizados no aprendizado de máquina tradicional nos seguintes aspectos: em aprendizado de máquina tradicional deseja-se melhorar uma medida de performance P e supõe-se que, reduzindo-se $J(\theta)$, P irá melhorar indiretamente, enquanto que em aprendizado profundo, o objetivo é reduzir J diretamente.

2.3.3.2.1 Propagação Reversa

A rede neural pode ser vista como um conjunto de funções compostas. Sendo assim, o algoritmo de propagação reversa (ou retropropagação do erro ou *backpropagation*) faz uso da Regra da Cadeia do cálculo diferencial para ajustar os parâmetros da rede [12, 205]. O gradiente da função de custo da rede neural é calculado com respeito aos parâmetros θ da mesma (em notação simbólica, $\nabla_{\theta} J(\theta)$). O vetor θ inclui todos os parâmetros $\mathbf{w}^{[l](i)}$ e $b^{[l](i)}$ de todas as camadas da rede. O gradiente deve ser calculado com respeito a cada um. O resultado de cada gradiente – um produto de derivadas parciais, conforme a Regra da Cadeia – é utilizado para se ajustar recursivamente os valores dos respectivos parâmetros.

O esquemático a seguir (Figura 23) ilustra o algoritmo de propagação reversa para uma rede de duas camadas:

Figura 23: Esquemático do algoritmo de propagação reversa para uma rede de duas camadas

Fonte: Curso Neural Networks and Deep Learning [14].

2.3.3.2.2 RMSProp

As taxas de aprendizado são alguns dos hiperparâmetros mais difíceis de se ajustar, pois têm impactos significativos na performance do modelo e as funções de custo têm diferentes sensibilidades para diferentes direções no espaço de parâmetros. Pesquisadores desenvolveram uma família de métodos adaptativos, que utiliza taxas de aprendizado separadas para cada parâmetro e ajusta estas taxas de forma automatizada e incremental, baseada em mini batches, durante o processo de treinamento. [12, 306]. Um destes métodos é o RMSProp. Este método pega um mini batch de m exemplos do conjunto de treinamento, acumula a média móvel dos quadrados dos gradientes e divide a taxa de aprendizado global pela raiz deste número. O vetor de parâmetros então é ajustado pelo gradiente da função de custo multiplicado por esta taxa ponderada [12, 307-308] [16].

O RMSProp foi projetado para ter boa performance com funções não convexas, contornando algumas limitações de outros algoritmos adaptativos, e tem se mostrado eficiente na prática, sendo um dos algoritmos de otimização mais utilizados em redes neurais na atualidade [12, 308].

2.4 A SEQUÊNCIA DE PROCEDIMENTOS DE APRENDIZADO DE MÁQUINA

De acordo com Chollet, todo trabalho de aprendizado de máquina segue determinados procedimentos [5]. Este trabalho segue o roteiro proposto pelo autor, que consiste nos seguintes passos:

- Definição do Problema de Aprendizado de Máquina;
- Escolha do Protocolo de Validação;
- Pré-Processamento e Engenharia de Atributos;
- Redução de Overfitting.

Tais passos serão discutidos na sequência.

2.4.1 Definição do Problema de Aprendizado de Máquina

Na definição do problema, busca-se determinar a tarefa do aprendizado de máquina: se é uma classificação binária, multiclasse, regressão escalar, vetorial, etc. Esta definição de base para a escolha da arquitetura do modelo e da função de perda, entre outras coisas.

A definição do problema depende da variável de saída. Tome-se um exemplo dado por Chollet [5]: classificação de resenhas de filmes. Neste caso, as variáveis de entrada são resenhas de filmes e as variáveis de saída são anotações de opinião. ‘Anotações de opinião’ são classificações humanas de acordo com a opinião expressa na resenha. Elas podem ser, por exemplo, ‘negativa’ e ‘positiva’, ou simplesmente 0 e 1, onde 0 pode significar uma opinião negativa e 1 uma positiva. Aqui, como a variável de saída é discreta, a tarefa a ser realizada é uma classificação. Além do mais, como ela só pode assumir dois valores, trata-se de uma classificação binária. Se a variável de saída fosse um valor contínuo, o problema seria uma regressão.

No exemplo acima foi assumido que o problema tem solução, mas nem sempre é o caso. Chollet cita outro exemplo: um algoritmo de aprendizado de máquina que recomenda roupas para a estação. Treina-se o modelo com dados do outono e se deseja obter uma recomendação para o inverno. Isto não funcionaria porque o modelo só pode fazer previsões para aquilo que ele foi treinado. Este tipo de problema é chamado de *problema não-estacionário*. Um problema não-estacionário é um problema que não pode ser resolvido. Pode-se, no máximo, contorná-lo, fazendo uma de duas coisas. A primeira

seria retrainar o modelo com novos dados para cada nova situação. Por exemplo, esperar o inverno chegar e retrainar o modelo “em tempo real”, na esperança de os novos dados sejam suficientes. Esta abordagem nem sempre é desejável porque os novos dados podem ser insuficientes e porque frequentemente se quer uma previsão com antecedência. A segunda solução seria usar uma escala de tempo grande o suficiente para que o problema se torne *estacionário*. No exemplo citado, dada a periodicidade das estações do ano, pode-se treinar o modelo com dados do ano passado. Assim pode-se obter uma recomendação para qualquer estação do ano a qualquer momento.

Uma terceira e última consideração sobre a definição do problema seria a escolha de um critério ou medida de sucesso. Com isto busca-se quantificar a performance do aprendizado de máquina. Em outras palavras, estimar o quanto ele está sendo bem-sucedido (ou não) na realização da tarefa proposta. Segundo Chollet, isto varia de caso a caso e está ligado ao objetivo maior do autor do modelo, mas há alguns casos típicos. Por exemplo, para um problema de classificação balanceado (em que todas as classes são igualmente prováveis), recomenda-se acurácia e área sob a curva característica de operação do receptor (curva ROC). Para classificação multirrotulo (*multilabel*), é recomendada a média das precisões médias (*mean average precision*). Outras métricas bastante utilizadas são: sensibilidade, especificidade, precisão, revocação (*recall*), etc. Enfim, há uma diversidade de métricas para se selecionar de acordo com o problema e o objetivo propostos.

2.4.2 Escolha do Protocolo de Validação

A validação é um método utilizado para se avaliar *overfitting*. Ocorre *overfitting* quando um modelo de aprendizado de máquina exibe um bom desempenho para dados usados no seu treinamento e mau desempenho para outros dados adicionais. Isto é diferente do caso do problema não-estacionário, discutido anteriormente. Naquele caso, tentava-se obter uma previsão sobre uma situação com a qual o modelo absolutamente não estava familiarizado. Neste caso, o modelo está familiarizado, mas de uma maneira viciada: os dados de treinamento não possuem informação suficiente sobre o problema real, seja porque o conjunto apresenta defeitos, ou é pequeno demais ou possui pouca variedade.

Segundo Chollet [5], os seguintes protocolos funcionam na maioria dos casos: *hold-out*, validação cruzada em K partes (*k-fold cross validation*) e validação cruzada em K partes iterada (*iterated k-fold validation*). No *hold-out*, uma parte do conjunto de treinamento é separada para validação (não confundir esta partição com o conjunto de

validação propriamente dito, reservado para depois) e treina-se o modelo com o restante. Este protocolo é recomendado quando se tem um conjunto de dados grande. Já na validação cruzada em K partes o conjunto de treinamento é dividido em K partições iguais, em que uma é utilizada para validação e as $K - 1$ restantes para treinamento. Este processo é realizado K vezes e a média das validações é utilizada como o escore final. Assim como no hold out, um conjunto de validação distinto deve ser reservado para depois. Este protocolo é recomendado quando o conjunto de dados é pequeno. Por fim, quando se tem poucos dados e se deseja a maior acurácia possível, recomenda-se a validação cruzada em K partes iterada. Este protocolo consiste em se realizar a validação cruzada em K partes várias vezes (*e.g.* P vezes), mas a cada iteração o conjunto é embaralhado. O escore final é então calculado utilizando-se as médias dos escores das P validações cruzadas (ao todo $P \times V$ modelos são treinados).

2.4.3 Pré-Processamento e Engenharia de Atributos

2.4.3.1 Pré-Processamento

Pode ser que os dados do conjunto precisem ser tratados computacionalmente antes de serem utilizados no modelo. Isto se faz necessário se, por exemplo, os dados estão representados em bases temporais diferentes (*e.g.* Tempo Universal Coordenado e horário de Brasília) ou faixas de valores diferentes (*e.g.* W e kW). Não é incomum ocorrer isto quando se misturam dados provenientes de fontes distintas.

Outra questão diz respeito a valores faltantes. Segundo Chollet [5], em aprendizado de máquina tradicional, é necessário eliminar uma linha ou mesmo uma coluna inteira de atributos se houver muitos dados faltantes. Já em se tratando de redes neurais, é seguro atribuir um valor nulo aos dados faltantes.

Para pré-processamento de dados em geral, o autor sugere o seguinte protocolo: 1) formatar os dados em forma de tensor; 2) colocar os valores em uma faixa pequena, entre $[-1, 1]$ ou $[0, 1]$; 3) se estiverem representados em faixas de valores diferentes, normalizá-los; 4) aplicar, se necessário, algum tipo de engenharia de atributos [5].

2.4.3.2 Engenharia de Atributos

Engenharia de atributos (*feature engineering*) consiste no “processo de usar o seu próprio conhecimento sobre os dados e sobre o algoritmo de aprendizado de máquina (...) para fazer o algoritmo funcionar melhor aplicando transformações em codificação

rígida (não aprendida) nos dados antes de usá-los no modelo” [2]. Em outras palavras, por meio de engenharia de atributos, o projetista usa seu conhecimento de alto nível dos dados para facilitar o trabalho da máquina. Por exemplo, um algoritmo que lê as horas de imagens de relógios: ao invés de se treinar o algoritmo diretamente com imagens digitalizadas, pode-se escrever um script que mapeia os ponteiros do relógio a coordenadas por meio de processamento digital de imagens, e treinar o algoritmo com as coordenadas como variáveis de entrada. Nos primórdios do aprendizado de máquina, engenharia de atributos era fundamental. Porém com a disponibilidade de processamento cada vez mais barata e o advento do aprendizado profundo, problemas que não poderiam dispensá-lo agora podem. Mesmo assim engenharia de atributos ainda é recomendável porque permite a algoritmos trabalharem de maneira mais eficiente e com menos dados [5].

2.4.4 Redução de Overfitting

Esta etapa trata da redução de overfitting através de treinamento de hiperparâmetros e regularização.

O primeiro passo é obter-se um modelo básico capaz de obter resultados estatisticamente significativos. Chollet [5] lembra que o aprendizado de máquina supõe duas hipóteses: 1) as saídas podem ser previstas com base nas entradas e 2) há dados suficientes. Se o modelo não conseguir resultados estatisticamente relevantes nesta etapa, isto significa que uma das ou ambas as condições não estão sendo satisfeitas, e uma revisão do trabalho se faz necessária. O que é estatisticamente significativo depende do tipo de problema e da métrica de sucesso escolhida: para uma classificação binária, uma acurácia maior do que 50% atende esse quesito, e assim por diante.

De posse de um modelo com poder estatístico, está-se quase pronto para o ajuste final de hiperparâmetros e regularização. Antes, porém, é preciso achar uma configuração que não apresente nem overfitting, nem underfitting. Para isto, será utilizado o protocolo de validação escolhido previamente, reservando-se o conjunto de validação para os ajustes finais.

Para se descobrir onde ocorre overfitting, deve-se “forçá-lo” a acontecer [5]. Isto é feito ajustando-se (hiper)parâmetros e monitorando-se as perdas de treinamento e de validação até estas começarem a se degradar. Enquanto as duas medidas decrescem juntas, diz-se que o modelo exibe underfitting; quando elas atingem um platô, chega-se na

²*process of using your own knowledge about the data and about the machine-learning algorithm (...) to make the algorithm work better by applying hardcoded (nonlearned) transformations to the data before it goes into the model (CHOLLET, 2018, p. 102).*

região desejada; e quando as curvas começarem a se separar – com a curva de treinamento descendo e a de validação subindo –, é sinal de que o modelo começou a exibir overfitting. Identificada a região onde o modelo não apresenta nem overfitting, nem underfitting, escolhe-se um ponto desta e prossegue-se para a etapa seguinte: ajuste de hiperparâmetros e regularização.

2.4.4.1 Ajuste de Hiperparâmetros

Hiperparâmetros são parâmetros que controlam o processo de aprendizado, e diferem dos demais parâmetros no seguinte: enquanto os demais parâmetros são ajustados automaticamente pelo próprio processo de treinamento (e.g. os pesos de uma rede neural), hiperparâmetros ficam a cargo do projetista (e.g. o número e tamanho de camadas de uma rede neural).

Alguns hiperparâmetros já começaram a ser ajustados na etapa anterior. Por isso mesmo foi reservado um conjunto de validação separado: assim como no conjunto de treinamento, também há risco de ocorrer overfitting no conjunto de validação. Cada vez que um hiperparâmetro é ajustado com base nele, um pouco de informação sobre o conjunto “vaza” para o modelo, e este vai progressivamente se ajustando àquele, o que pode levar, no limite, ao overfitting. Porém, tendo realizado validações apenas em subdivisões do conjunto de treinamento até agora, preservou-se um conjunto de validação intacto para os ajustes finais.

2.4.4.2 Regularização

De acordo com Chollet [5], a melhor solução para se prevenir overfitting é adquirir mais dados. A segunda melhor solução é forçar o algoritmo a ignorar aspectos irrelevantes do conjunto e focar apenas nos aspectos mais proeminentes, aumentando assim o nível de generalidade. É isto que se busca realizar com a regularização. Algumas técnicas de regularização são citadas a seguir.

- Reduzir o tamanho da rede, reduzindo o número de parâmetros treináveis (*learnable*): isto significa substituir os dados brutos com representações comprimidas dos mesmos. Por exemplo, ao invés de se usar uma imagem de alta resolução, aplica-se uma camada convolucional para extrair apenas as features mais relevantes. Não existe uma regra formal para este procedimento; recomenda-se começar com um modelo com um mínimo de camadas e elementos por camada e aumentar a complexidade da rede progressivamente, sempre monitorando as perdas de validação.

- Adicionar regularização ponderada: com este método busca-se simplificar o modelo atribuindo pesos aos parâmetros: maiores para os mais importantes, e menores – tendendo a zero – para os menos. Isto é realizado adicionando-se à função perda do modelo um custo associado aos coeficientes de pesos. Por exemplo: regularização ℓ_1 : o custo é proporcional ao módulo dos coeficientes de pesos (norma L1); regularização ℓ_2 : o custo é proporcional ao quadrado dos coeficientes de pesos (norma L2).
- Adicionar *dropout*: literalmente “jogar fora” (*drop out*) atributos de camadas aleatoriamente durante o treinamento – atribuindo-lhes valor zero –, segundo uma taxa definida (geralmente de 0.2 a 0.5, representando a razão entre a quantidade de features zerados e o total). Na etapa de teste, porém, ao invés de serem zeradas, as features são multiplicadas pela taxa de dropout escolhida para compensar o excesso de unidades ativas, comparado ao treinamento.

Cabe ao projetista decidir que técnicas de regulação e hiperparâmetros utilizar. Esta etapa consiste em boa parte de tentativa e erro. A cada modificação deve-se testar o modelo com o conjunto de validação, levando em consideração o overfitting. Quando a performance do modelo estiver satisfatória, treina-se o modelo uma última vez, combinando-se os conjuntos de treinamento e validação, e testa-se no conjunto de teste. Se a performance no conjunto de teste for significativamente pior do que no conjunto de validação, isto pode significar que o processo de validação foi falho ou apresentou overfitting no conjunto de validação. Neste caso, recomenda-se a utilização de um protocolo de validação mais confiável, como validação cruzada em K partes iterada.

2.5 REVISÃO DO ESTADO DA ARTE

Para a revisão do estado da arte foram selecionados alguns artigos da biblioteca digital do IEEE, a IEEE Xplore. Todos os artigos pesquisados são recentes e tratam do mesmo problema de predição de geração fotovoltaica, utilizando abordagens diferentes. A revisão procurou levantar os seguintes aspectos: que abordagem de aprendizado de máquina foi utilizada; o que foi incluído no conjunto de dados; qual foi o horizonte temporal da previsão e quais foram os resultados obtidos. A seguir, é feito um breve resumo de cada artigo na ordem em que eles aparecem nas referências:

Ogawa *et al.* [17] propõem o uso de uma rede neural evolutiva (*evolutionary neural network*). Trata-se de um método baseado em redes neurais que utiliza máquina de Boltzmann restrita (*restricted Boltzmann machine* – RBM) e otimização de brainstorm predador-presa de computação evolutiva (*predator-prey brain storm optimization of*

evolutionary computation). O modelo utiliza dados de uma unidade de geração fotovoltaica no Japão. A previsão é para 30 minutos depois. O método proposto foi testado com dados reais e comparado com métodos convencionais. Os resultados da simulação indicam que o método proposto obteve resultados melhores em termos de erros máximo e médio e desvio padrão do que os métodos tradicionais.

Kaffash *et al.* [18] utilizaram uma *ensemble* de redes neurais artificiais (*artificial neural networks* – ANN) e máquinas de vetores de suporte (*support vector machine* – SVM). Foram coletados dados de geração em intervalos de uma em uma hora. As features foram selecionadas através de um processo chamado seleção de características de florestas aleatórias (*random forest feature selection*). A previsão foi para o dia seguinte (com dados apenas do dia anterior). Os dados são provenientes de uma instalação fotovoltaica em Gent, na Bélgica (EnergyVille-1). Os resultados da simulação mostram que, separadamente, a abordagem baseada em SVM teve melhor desempenho do que a baseada em ANN na maior parte do tempo, mas em alguns casos as redes neurais obtiveram melhores resultados. O ensemble das duas, por sua vez, obteve um resultado mais balanceado, e foi preferido pelos autores.

Teo *et al.* [2] utilizaram um algoritmo de redes neurais artificiais chamado de aprendizado de máquina extremo (*extreme machine learning*). Os dados foram adquiridos de 8 de junho a 6 de julho de 2014 e a previsão foi para cinco dias (2 a 6 de julho) daquele ano. As variáveis de entrada consideradas foram: temperatura ambiente [°C], temperatura do painel [°C], energia diária acumulada [J], irradiância [Wm^{-2}]. Já a variável de saída é a potência gerada [W]. A função de custo escolhida foi a raiz quadrada do erro médio (*root-mean-square error* – RMSE). Diferentes modelos foram testados para se encontrar o que teve o melhor desempenho, bem como diferentes tamanhos de conjunto de treinamento e número de variáveis de entrada. O resultado da simulação foi considerado condizente com os dados reais e foram sugeridos futuros experimentos.

Theocharides *et al.* [19] testaram várias abordagens – separadamente, não em ensemble como M. Kaffash *et al.* [18] – com vários hiperparâmetros e features e compararam seus resultados. As abordagens utilizadas foram ANN, SVM e árvores de regressão (*regression trees* – RT). O conjunto de dados foi coletado durante um ano de um sistema fotovoltaico conectado à rede de uma universidade no Chipre. As métricas de sucesso utilizadas foram erro percentual absoluto médio (*mean absolute percentage error* – MAPE) e erro quadrático médio normalizado (*normalised root mean square error* – nRMSE). Foi considerado que a abordagem baseada em redes neurais obteve o melhor desempenho.

Jiang *et al.* [20] utilizam um modelo de aprendizado de máquina extremo (*extreme machine learning*) baseado em dias similares, usando o conceito de similaridade melhorada (*improved similarity*). A ideia deste conceito é descobrir os fatores meteorológicos que mais influenciam a mudança de geração fotovoltaica usando correlação de Pearson. Os fatores dominantes dos dados de geração diária são selecionados e dispostos em um vetor de features. A similaridade melhorada é então estabelecida com base no vetor de características. Por meio deste processo, a predição é realizada com base nos dias semelhantes. Os resultados experimentais mostraram uma acurácia de predição maior e um tempo de computação menor do que os métodos tradicionais, além de o algoritmo ser simples e de baixo custo.

2.5.1 Discussão sobre os Artigos Pesquisados

Os autores utilizam uma série de abordagens de aprendizado de máquina, desde as mais tradicionais [19] até mais avançadas [17], bem como um ensemble de abordagens diferentes [18]. A composição do conjunto de dados não foi detalhada na maioria dos casos – com os autores limitando-se a especificar a origem dos dados –, com exceção de Teo *et al.* [2], que ofereceram uma lista detalhada.

Os horizontes temporais das previsões foram desde meses [2], um dia [18], ou até 30 minutos [17] de antecedência. Todos relataram resultados promissores, com diferentes graus de sucesso, com destaque para Ogawa *et al.* [17] e Jiang *et al.* [20], que relataram resultados superiores aos obtidos com abordagens de aprendizado de máquina tradicionais.

3 DESENVOLVIMENTO

Neste capítulo, é apresentado o conjunto de dados e suas peculiaridades. Em seguida, é relatada a implementação do modelo de aprendizado de máquina.

3.1 O CONJUNTO DE DADOS

O conjunto de dados utilizado neste trabalho foi montado com dados de duas fontes distintas, uma para os dados ambientais e outros para os dados de geração de energia. O motivo disso foi que não foi possível encontrar os dois tipos de dados em um só lugar. Os dados ambientais foram retirados da base de dados da rede SONDA, e os dados de geração, do SUNNY Portal.

3.1.1 Rede SONDA

A rede SONDA (Sistema de Organização Nacional de Dados Ambientais) é uma iniciativa do Instituto Nacional de Pesquisas Espaciais (INPE), que tem por objetivo a “implementação de infra-estrutura física e de recursos humanos para levantar e melhorar a base de dados dos recursos de energia solar e eólica do Brasil” [21].

Os dados coletados são submetidos a um rigoroso processo de avaliação em que as medições são checadas para adequação aos limites físicos, consistência entre sensores diferentes e consistência histórica entre as componentes direta, difusa e global em comparação com valores estimados por modelos. Posteriormente, os dados ainda são submetidos a um processo de validação seguindo um protocolo internacional, em que medições suspeitas são sinalizadas, sem contudo alterá-las.

A base de dados é dividida em dados ambientais e dados anemométricos. Os dados ambientais, por sua vez, compõem-se de dados radiométricos e meteorológicos. A periodicidade dos dados ambientais é de 1 em 1 minuto, e dos anemométricos, de 10 em 10 minutos. Neste trabalho apenas os dados ambientais são relevantes. A rede conta com estações de medição próprias e parceiras em vários locais do Brasil, incluindo uma em Florianópolis. As estações são do tipo solarimétricas (S), anemométricas (A) ou solarimétricas e anemométricas (SA). As variáveis medidas dependem do tipo de estação e dos sensores existentes em cada estação, mas por padrão a rede disponibiliza planilhas

com 17 colunas para dados ambientais e 11 colunas para dados anemométricos. Se um dado de uma variável for faltante, a célula é preenchida com a sigla N/A (*not available*); se a estação não possui o sensor para medição de uma variável, a coluna é preenchida com a sigla N/S (*no sensor*).

A estação localizada em Florianópolis (FLN - 01) é uma estação parceira e do tipo S. Ela fica na latitude 27° 36' 6,1" S e longitude 48° 31' 4,2" O e altitude 31 m. As variáveis medidas pela estação de Florianópolis e suas correspondentes unidades de medida são mostradas na Tabela 1:

Tabela 1: Tabela de dados ambientais da rede SONDA

Dados Radiométricos	
Radiação Global Horizontal	Wm ⁻²
Radiação Direta Normal	Wm ⁻²
Radiação Difusa	Wm ⁻²
Radiação de Onda Longa Descendente	Wm ⁻²
Dados Meteorológicos	
Temperatura do Ar na Superfície	°C
Umidade Relativa do Ar	%
Pressão do Ar	mbar
Precipitação de Chuva	mm
Velocidade do Vento a 10 m	ms ⁻¹
Direção do Vento a 10 m	° (horário)

Fonte: Elaborada pelo autor.

A título de comparação, Teo *et al.* [2] utilizam os seguintes dados para treinar seu modelo de aprendizado de máquina:

Tabela 2: Conjunto de dados Teo et al. (2016)

Temperatura Ambiente	°C
Temperatura do Painel	°C
Energia Diária Acumulada	J
Irradiância	Wm ⁻²
Potência	W

Fonte: Elaborada pelo autor.

Algumas diferenças entre os dois conjuntos são aparentes: o conjunto da rede SONDA tem mais variáveis, mas nem todas são relevantes para geração fotovoltaica; o conjunto de Teo *et al.* possui informações de temperatura do painel e ambiente, a SONDA apenas ambiente; e os dados de irradiância na rede SONDA são discriminados em global horizontal, direta normal e difusa (vide Seção 2.1.1 do Capítulo 2).

Observa-se que a irradiância, no conjunto de dados de Teo *et al.*, corresponde à soma das componentes direta e difusa da radiação. Finalmente, nota-se que o conjunto de Teo *et al.* possui dados de energia e potência enquanto que os da rede SONDA não. Isto é esperado, visto que esta se trata de uma estação meteorológica, não uma planta fotovoltaica.

3.1.2 SUNNY Portal

Para se obter os dados de energia e potência recorreu-se a uma fonte diferente, o SUNNY Portal. O SUNNY Portal [22] é um site que disponibiliza dados de medição de sistemas fotovoltaicos em todo o mundo, bem como dados de redução de emissões de CO² e arquivagem gratuita de dados, com criação de páginas individuais para cada sistema. Em Florianópolis, sete sistemas constam na base de dados do site. Os nomes dos sistemas (*ipsis litteris*) e suas potências instaladas estão dispostos na Tabela 3:

Tabela 3: Sistemas fotovoltaicos de Florianópolis constantes na base de dados do SUNNY Portal

Nome do Sistema	Potência (kWp)
Colégio Aplicação - UFSC	1,99
Diko B - Florianópolis	3,30
ELETROSUL - PLANTA PILOTO	11,97
FOTOVOLTAICA - UFSC - 2kWp	2,02
Fotovoltaica Rotária do Brasil	20,55
Sede	12,28
Vereda Tropical	49,64

Fonte: Elaborada pelo autor.

Por padrão, os dados de geração são divididos em diário, mensal, anual e total: os dados diários são medições de potência média de 15 em 15 minutos; os dados mensais

são medições da energia diárias; os dados anuais são medições de energia mensais; e o total são medições de energia anuais desde o início do funcionamento do sistema.

3.1.3 Discussão sobre o Conjunto de Dados

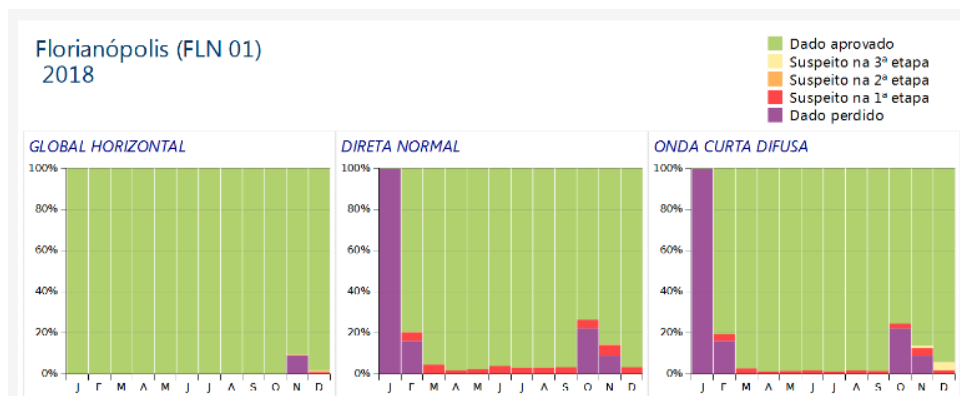
Não se verificou consistência na disponibilidade e qualidade dos dados do SUNNY Portal: os sistemas “Colégio Aplicação - UFSC” e “Diko B - Florianópolis” possuem apenas perfis cadastrados, mas não disponibilizaram outros dados; “Vereda Tropical” possui apenas dados de 2020 e “ELETROSUL - PLANTA PILOTO” possui apenas medições mensais. Dos três restantes, “Fotovoltaica - Ufsc” não fornece dados desde 2015, e todos os sistemas contêm lacunas de dias, meses e até anos nas suas medições.

Já a rede SONDA (estação de Florianópolis) teve as atividades interrompidas de março a agosto de 2013 e parou de fornecer dados a partir de maio de 2019. Somado a isto, esta base de dados possui meses e – dependendo da variável medida – anos inteiros com dados marcados como perdidos ou suspeitos.

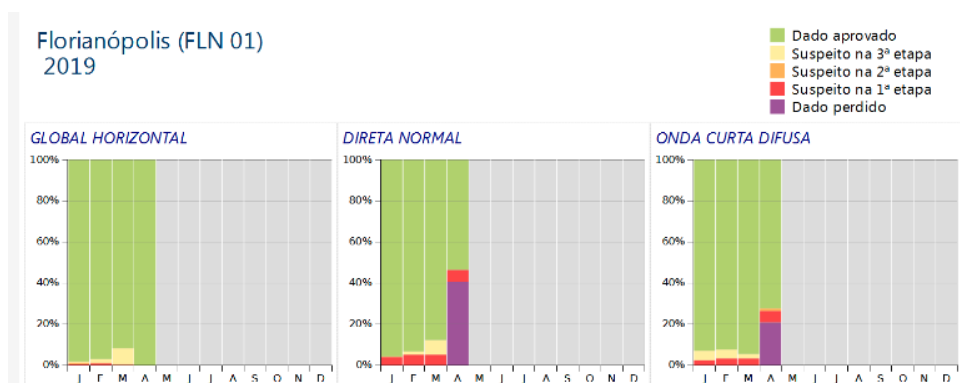
Nessas circunstâncias, o melhor que se pode fazer é encontrar um período em que ambos os conjuntos de dados apresentem a maior integridade dos dados possível. O sistema do SUNNY Portal que apresentou o melhor período coincidente com o melhor período da rede SONDA é o sistema “Sede” de janeiro de 2018 a abril de 2019.

Escolhido o período com os dados mais completos nos dois conjuntos, porém, constata-se que este ainda é aquém do ideal. As Figuras 24-28 foram escolhidas para ilustrar alguns problemas com os dados de ambos os conjuntos.

A Figura 24 e 25 são da rede SONDA. Elas mostram a porcentagem mensal de dados aprovados pelo processo de avaliação (em verde), a porcentagem dos dados marcados como suspeitos em cada uma das três etapas de validação (em amarelo, laranja e vermelho) e a porcentagem de dados perdidos (em roxo) de algumas variáveis nos anos de 2018 e 2019. Nota-se, na Figura 24, que 100% dos dados de janeiro foram perdidos, e cerca de 20% dos dados de fevereiro, outubro e novembro foram classificados como suspeitos ou perdidos em 2018. Já a Figura 25 mostra que a porcentagem de dados suspeitos ou perdidos em uma variável chega a quase 50% em abril de 2019, e a partir de maio não há mais dados.

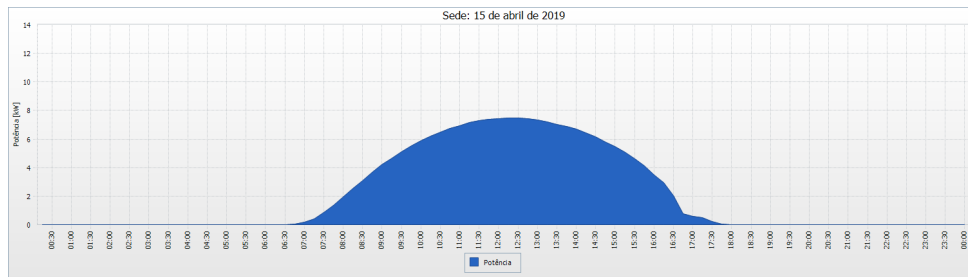
Figura 24: Dados da rede SONDA do ano de 2018

Fonte: Rede SONDA [21].

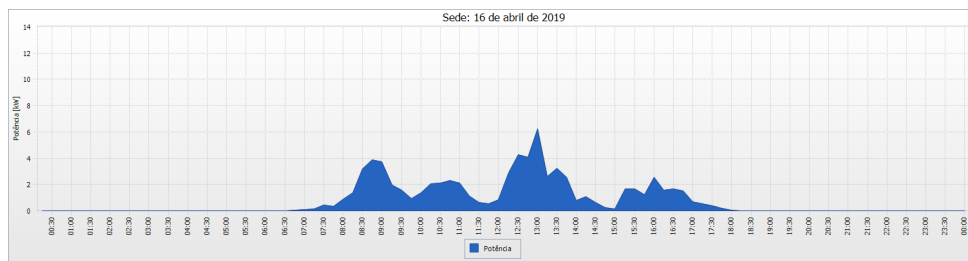
Figura 25: Dados da rede SONDA do ano de 2019

Fonte: Rede SONDA [21].

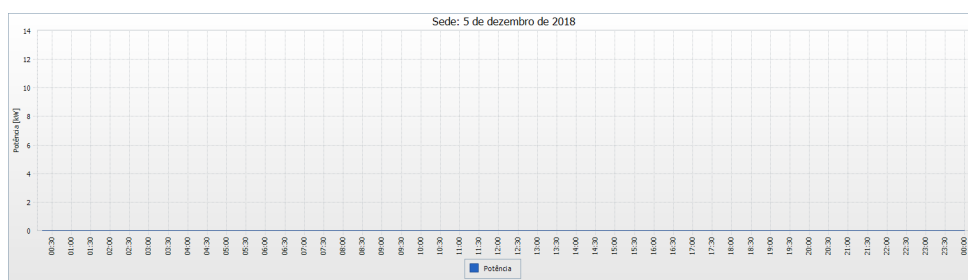
As próximas figuras são do portal SUNNY. A Figura 26 mostra a potência gerada pelo sistema "Sede" no dia 15 de abril de 2018. O formato do gráfico é exatamente o que se espera: uma curva em forma de sino e com pico em torno do meio-dia. A Figura 27 mostra o gráfico do dia seguinte, 16 de abril de 2018: este gráfico, ao contrário do primeiro, apresenta uma série de deformações. Estas provavelmente se devem a condições climáticas desfavoráveis (conforme discutido no Capítulo 2). Finalmente, a Figura 28 mostra o gráfico do dia 5 de dezembro de 2018. Neste dia todos os valores de potência são nulos: os dados nem sequer foram registrados, possivelmente devido a problemas de conexão.

Figura 26: Geração do Sistema “Sede” de 15 de abril de 2018

Fonte: Portal SUNNY [22].

Figura 27: Geração do Sistema “Sede” de 16 de abril de 2018

Fonte: Portal SUNNY [22].

Figura 28: Geração do Sistema “Sede” de 5 de dezembro de 2018

Fonte: Portal SUNNY [22].

Nota-se, portanto, que, mesmo havendo sido selecionado um período favorável, ambas as bases de dados ainda exibem dados faltantes e defeituosos.

3.1.4 Perfil do Sistema Sede

A Tabela 4 reúne os dados do sistema Sede no site SUNNY Portal. Nota-se que, fora o endereço e os dados técnicos, nenhum detalhe adicional é fornecido sobre o sistema. Isto estaria em contradição com a proposta do trabalho de se realizar uma predição sobre um sistema fotovoltaico *conhecido*.

Tabela 4: Dados do sistema Sede

Local de instalação	Florianópolis, SC, Brasil, CEP 88080220
Colocação em serviço	02/08/2017
Potência do sistema	12,280 kWp
Produção anual	Aprox. 176 832 kWh (14 400 kWh/kWp)
Redução das emissões de CO ²	Aprox. 123,8 toneladas por ano
Comunicação	SMA Webconnect
Inversor	Sunny Tripower 12000TL-20

Fonte: Elaborada pelo autor.

Contudo, uma matéria de jornal de 11 de agosto de 2017 [23] – mesmo mês que a colocação em serviço do sistema considerado – tem o seguinte título: “Empresa em Florianópolis deixará de emitir 200 toneladas de CO² com painéis solares: A usina para a geração de energia fotovoltaica foi instalada em três dias no telhado da sede da organização.” A matéria menciona que o sistema instalado tem capacidade de geração de 11,7 kWp. Ignorando as 200 toneladas de CO² anunciadas, a potência de 11,7 kWp está razoavelmente próxima da de 12,280 kWp da Tabela 4. Finalmente, uma rápida busca na internet revela que a sede da empresa citada na matéria possui o mesmo CEP que o sistema “Sede” do portal SUNNY, confirmando a sua identidade. Outras informações, não constantes no portal, são que o sistema possui 36 painéis solares e atende 30% da demanda mensal por energia da empresa.

3.2 IMPLEMENTAÇÃO

Nesta seção é apresentada a implementação dos assuntos propostos na Seção 2.4.

3.2.1 Definição do Problema

Na Seção 2.4.1 foi comentado que a definição do problema passa pela descrição das variáveis de entrada e saída do modelo de aprendizado de máquina. A escolha das variáveis de entrada dependeria de um estudo do funcionamento do sistema fotovoltaico com o objetivo de determinar as grandezas físicas que influenciam a eficiência do sistema. Na Seção 2.1, tal estudo é apresentado. Além disso, outra questão a ser levada em consideração é a disponibilidade dos dados. Após uma pesquisa na internet, foi possível encontrar bases de dados adequadas tanto para as variáveis de entrada quanto as de saída (vide Seção 3.1). As variáveis de entrada são dados ambientais e a de saída é a potência gerada por uma planta fotovoltaica.

Tendo definido as variáveis de entrada e saída, é possível afirmar que o problema se trata de uma regressão. Isto porque a variável de saída não se trata de uma classe ou rótulo, mas de uma grandeza escalar.

Outra questão a ser respondida é se o problema é estacionário. Conforme visto na Seção 2.3.1, um problema não-estacionário não tem solução. Felizmente, este não é o caso. O problema proposto neste trabalho é, em um aspecto, semelhante a um exemplo discutido naquela Seção, o de recomendação de roupas para a estação. Sabe-se que a geração fotovoltaica depende de fatores ambientais, e estes são influenciados pelas estações do ano. Sabe-se ainda que as estações são periódicas. Logo, se o modelo for treinado com dados de um período de um ano anterior, espera-se (todo o mais constante) poder prever a geração para o mesmo período do ano seguinte, tornando-se o problema estacionário.

Contudo, uma análise mais refinada revela que a semelhança com o problema das roupas da estação é apenas superficial. Naquele problema, a diferença entre os dados de cada estação é qualitativa – refletindo questões culturais, etc. –, não sendo possível a máquina deduzí-los sem prévio contato com eles. No problema da geração fotovoltaica, contudo, a diferença é apenas quantitativa: do ponto de vista dos sensores – e consequentemente do algoritmo – o que muda entre uma estação e outra é tão somente a incidência maior ou menor de luz, a temperatura maior ou menor, etc., ou seja, mais ou menos do mesmo. Conclui-se, portanto, que o problema é estacionário.

Esta conclusão está de acordo com os artigos apresentados na Seção 2.4. Em nenhum dos artigos pesquisados a questão da não estacionariedade foi considerada, e nenhum precisou recorrer à periodicidade das estações. Pelo contrário, os algoritmos foram capazes de realizar previsões com meses, dias e, em um caso extremo, 30 minutos de antecedência.

A última questão a ser definida nesta etapa é a métrica de sucesso. Por se tratar de uma regressão, a principal métrica utilizada para a avaliação da performance é o erro quadrático médio (EQM). Contudo, além do EQM, também foram monitorados o erro absoluto médio (EAM) e o valor da função de perda (*loss*), sendo este último simplesmente o EQM, quando o modelo não possui regularização, e o EQM penalizado com um custo, quando possui regularização.

O método de otimização escolhido é o RMSProp, conforme apresentado na Seção 2.3.3.2.

3.2.2 Validação

A validação é realizada monitorando-se a perda no conjunto de validação, usando o EQM como função de perda.

3.2.3 Pré-Processamento e Feature Engineering

Conforme descrito na Seção 3.1, o conjunto de dados foi obtido combinando-se 16 meses de dados – janeiro de 2018 a abril de 2019 – de duas fontes distintas, a rede SONDA e o SUNNY Portal. Porém, antes de combiná-las é necessário resolver alguns problemas de compatibilidade por meio de pré-processamento.

O primeiro problema é a diferença nas taxas de amostragem entre as fontes. Os dados da SONDA são amostrados de 1 em 1 minuto e os do SUNNY, de 15 em 15. Para deixar o SUNNY na mesma taxa que o SONDA, foi realizada uma interpolação. O próximo problema identificado é uma disparidade na base temporal. A SONDA utiliza o sistema do Tempo Universal Coordenado (UTC), e o SUNNY, a hora legal do local de origem dos dados: no caso de Florianópolis, o horário de Brasília, que corresponde a três horas atrasadas em relação ao UTC. Além disso, nos anos de 2018 e começo de 2019 ainda vigorava no Brasil o horário de verão, o que significa que de outubro a fevereiro a hora legal se encontrava adiantada em 60 minutos, ficando duas horas atrasada com relação ao UTC ao invés de três. Assim, foi necessário identificar o período exato do horário de verão conforme a legislação da época e fazer os ajustes correspondentes [24].

O resultado deste pré-processamento é uma matriz com 698220 linhas e 11

colunas (Figura 29). Nessa matriz, as 10 primeiras colunas são dados de entrada – correspondendo aos atributos da Tabela 1 – e a última coluna, os dados de saída, contendo as leituras de potência do sistema Sede.

Figura 29: Conjunto de dados

date	glo_avg	dir_avg	diff_avg	lw_avg	tp_sfc	humid	press	rain	ws_10m	wd_10m	pwr
2018-01-01 00:00:00	-1.105	0.0	0.0	422.9	24.12	0.70	1005.22	0.0	0.094	148.2	0.0
2018-01-01 00:01:00	-1.183	0.0	0.0	421.9	24.08	0.65	1005.22	0.0	0.181	213.2	0.0
2018-01-01 00:02:00	-1.209	0.0	0.0	421.0	24.26	0.69	1005.16	0.0	0.086	190.0	0.0
2018-01-01 00:03:00	-1.238	0.0	0.0	419.7	24.18	0.70	1005.19	0.0	0.093	181.3	0.0
2018-01-01 00:04:00	-1.305	0.0	0.0	418.4	24.16	0.70	1005.19	0.0	0.000	0.0	0.0
...
2019-04-30 20:55:00	-1.985	0.0	0.0	377.3	21.43	81.70	1010.85	0.0	1.274	25.4	0.0
2019-04-30 20:56:00	-1.982	0.0	0.0	376.9	21.44	81.50	1010.88	0.0	1.563	16.3	0.0
2019-04-30 20:57:00	-1.889	0.0	0.0	376.7	21.44	81.90	1010.88	0.0	1.025	24.3	0.0
2019-04-30 20:58:00	-1.871	0.0	0.0	376.6	21.47	82.00	1010.85	0.0	0.387	51.4	0.0
2019-04-30 20:59:00	-1.893	0.0	0.0	376.5	21.44	81.80	1010.88	0.0	0.083	45.7	0.0

Fonte: Elaborada pelo autor.

Legenda dos atributos e unidades de medida:

- glo_avg: Radiação Global Horizontal (Wm^{-2})
- dir_avg: Radiação Direta Normal (Wm^{-2})
- diff_avg: Radiação Difusa (Wm^{-2})
- lw_avg: Radiação de Onda Longa Descendente (Wm^{-2})
- tp_sfc: Temperatura do Ar na Superfície ($^{\circ}\text{C}$)
- humid: Umidade Relativa do Ar (%)
- press: Pressão do Ar (mbar)
- rain: Precipitação de Chuva (mm)
- ws_10m: Velocidade do Vento a 10 m (ms^{-1})

- wd_10m: Direção do Vento a 10 (°)
- pwr: Potência Elétrica (W)

Foi implementado um programa em Python para automatizar o processo de formatação do conjunto de dados e escrever o resultado em uma planilha, conforme mostrado na Figura 24.

Os valores faltantes foram substituídos por zero. Deve-se ainda normalizar os dados de entrada antes de utilizá-los no modelo. Tal normalização corresponde a subtrair a média e dividir pelo desvio padrão cada amostra do conjunto.

Em matéria de feature engineering, foram removidos os atributos que não afetam diretamente a geração fotovoltaica. Sabe-se que velocidade e direção do vento, humidade do ar e pressão atmosférica não influenciam no rendimento do sistema (vide Seção 2.2). Experimentos preliminares com um modelo baseline determinaram que, de fato, a remoção destes atributos resulta em um desempenho melhor do aprendizado de máquina. Assim sendo, estes dados foram eliminados do conjunto, reduzindo o número de atributos de 10 para 6, conforme apresentado na Figura 30.

	glo_avg	dir_avg	diff_avg	lw_avg	tp_sfc	rain	pwr
date							
2018-01-01 00:00:00	-1.105	0.0	0.0	422.9	24.12	0.0	0.0
2018-01-01 00:01:00	-1.183	0.0	0.0	421.9	24.08	0.0	0.0
2018-01-01 00:02:00	-1.209	0.0	0.0	421.0	24.26	0.0	0.0
2018-01-01 00:03:00	-1.238	0.0	0.0	419.7	24.18	0.0	0.0
2018-01-01 00:04:00	-1.305	0.0	0.0	418.4	24.16	0.0	0.0
...
2019-04-30 20:55:00	-1.985	0.0	0.0	377.3	21.43	0.0	0.0
2019-04-30 20:56:00	-1.982	0.0	0.0	376.9	21.44	0.0	0.0
2019-04-30 20:57:00	-1.889	0.0	0.0	376.7	21.44	0.0	0.0
2019-04-30 20:58:00	-1.871	0.0	0.0	376.6	21.47	0.0	0.0
2019-04-30 20:59:00	-1.893	0.0	0.0	376.5	21.44	0.0	0.0

Figura 30: Conjunto de dados após feature engineering.

3.2.4 Redução de Overfitting

Do conjunto de dados total, foram considerados 2 meses, subdivididos em três subconjuntos. A escolha de 2 meses (de um total de 16) foi devida a uma questão de economia, e a escolha dos meses específicos (março e abril) foi em razão da confiabilidade dos dados nestes meses. Os três subconjuntos são os seguintes:

- conjunto de treinamento: 31 dias (01/03/2018 a 31/03/2018) ou 44640 pontos de dados;
- conjunto de validação: 15 dias (01/04/2018 a 15/04/2018) ou 21600 pontos de dados;
- conjunto de teste: 15 dias (16/04/2018 a 30/04/2018) ou 21600 pontos de dados.

Conforme apresentado na Seção 2.4.4, para o controle de overfitting, é recomendável iniciar com um modelo básico e ir aumentando sua complexidade progressivamente, sempre monitorando a perda no conjunto de validação.

4 EXPERIMENTOS E RESULTADOS

Este capítulo é dividido em três seções. A primeira trata do problema da previsão de geração com dados instantâneos (modelagem). Já na segunda e na terceira são realizadas previsões com antecedência de 30 minutos e 1 hora.

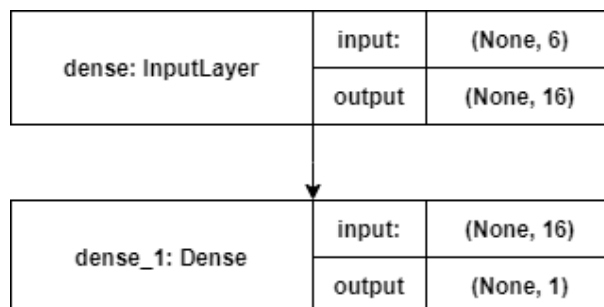
4.1 MODELAGEM

A seguir são relatados experimentos realizados com diferentes modelos de redes neurais. Nesses, salvo especificado em contrário, a função de ativação é ReLU e o método de otimização é RMSProp com taxa de aprendizado de 0,0001.

O número ideal de épocas de treinamento de cada modelo é determinado programando-o para ser treinado por 1000 épocas, com 20% do conjunto de treinamento separado para validação. Por meio de uma função de *callback*, o processo é interrompido quando a perda de validação não melhorar por 10 épocas consecutivas segundo uma tolerância $\text{delta_min} = 0$.

Partiu-se de um modelo básico ou *baseline* (Modelo A) com duas camadas densas: a primeira com 16 unidades e a segunda com 1, totalizando 129 parâmetros treináveis. A arquitetura do Modelo A é ilustrada na Figura 31. Uma vez treinado, avaliou-se o Modelo A no conjunto de validação. Os resultados são mostrados na Tabela 5.

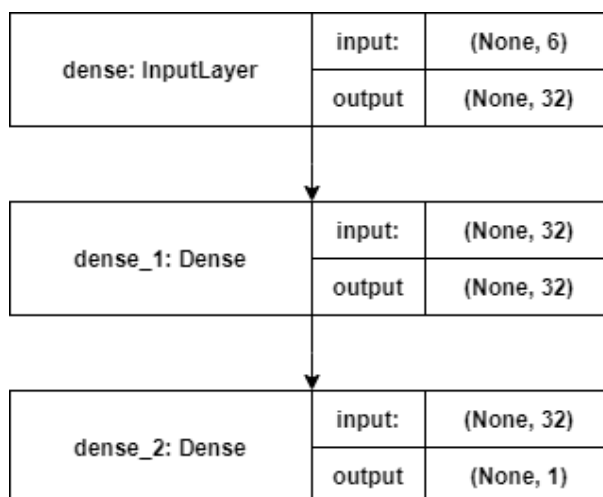
Figura 31: Diagrama de blocos do Modelo A



Fonte: Elaborada pelo autor.

Em seguida, criou-se um novo modelo (Modelo B) com três camadas densas: as duas primeiras com 32 unidades e a terceira com 1, totalizando 1313 parâmetros treináveis. O Modelo B é ilustrado na Figura 32.

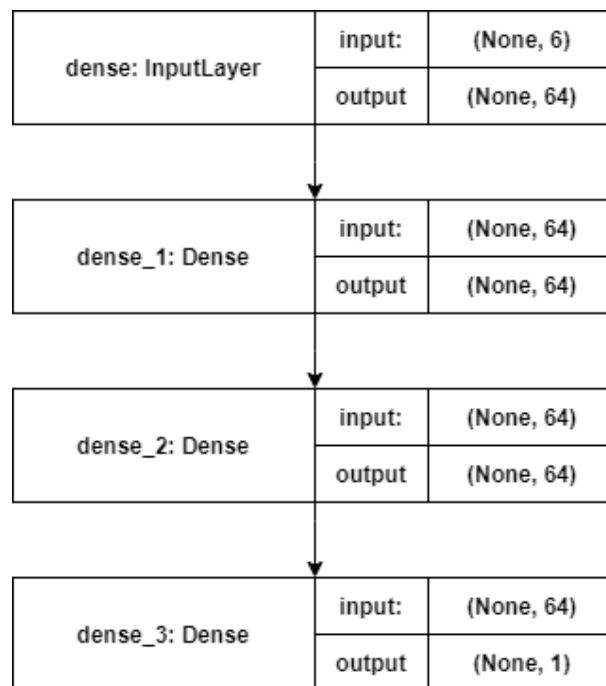
Figura 32: Diagrama de blocos do Modelo B



Fonte: Elaborada pelo autor.

O Modelo B foi treinado e avaliado no conjunto de validação. Os resultados são mostrados na Tabela 5.

Nota-se que o Modelo B obteve resultados melhores do que o A. Assim, criou-se um novo modelo (Modelo C) com quatro camadas densas: as três primeiras com 64 unidades e a quarta com 1, totalizando 8833 parâmetros treináveis. O Modelo C é ilustrado na Figura 33. O Modelo C foi treinado e avaliado da mesma forma que os anteriores. Sua performance é mostrada na Tabela 5.

Figura 33: Diagrama de blocos do Modelo C

Fonte: Elaborada pelo autor.

Considerou-se que a melhora do Modelo C com relação ao B foi muito pequena, não justificando o correspondente aumento do número de parâmetros treináveis. Portanto, o Modelo B foi escolhido para a etapa seguinte, de regularização.

Para a regularização, adicionou-se regularização ℓ_2 e uma camada dropout após cada camada oculta (isto é, todas menos a última) do Modelo B.

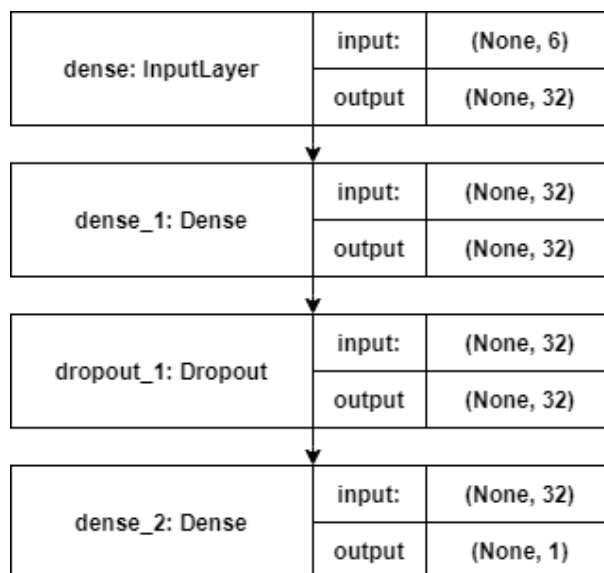
Já os ajustes dos valores das taxas de regularização ℓ_2 e dropout de cada camada, bem como da taxa de aprendizado do otimizador, foram realizados de modo automatizado utilizando-se o Keras Tuner [25].

Os valores finais destes hiperparâmetros são os seguintes:

- camada densa (input)
 - regularização ℓ_2 : 0,009662;
- dropout: 0;
- camada densa 1
 - regularização ℓ_2 : 0,005849;
- dropout 1: 0,1;
- taxa de aprendizado do RMSProp: 0,000412.

Observa-se que a busca de hiperparâmetros retornou uma taxa zero para a primeira camada de dropout, o que equivale a não considerar esta camada. Logo, ela foi eliminada

Assim sendo, o modelo final (Modelo B com regularização) é mostrado na Figura 34.

Figura 34: Diagrama em blocos do modelo final

Fonte: Elaborada pelo autor.

O modelo final foi treinado da mesma forma que os modelos anteriores, obtendo os resultados na Tabela 5.

Tabela 5: Avaliação dos modelos no conjunto de validação

Modelo	loss	EAM	EQM
Modelo A	0,5496	0,3676	0,5496
Modelo B	0,4984	0,2875	0,4984
Modelo C	0,4558	0,2759	0,4558
Modelo Final	0,5102	0,2768	0,4605

Fonte: Elaborada pelo autor.

Nota-se que neste caso a métrica loss exibiu um valor um pouco maior do que o EQM, quando em todos os outros experimentos as duas métricas tinham valores iguais. Isto se deve à regularização ℓ_2 , que penaliza a função de perda. Porém, considerando-se apenas a métrica EQM, o Modelo B com regularização obteve o melhor desempenho.

O modelo final foi retreinado com os conjuntos de treinamento e validação

combinados e avaliado no conjunto de teste. Para comparação, o mesmo procedimento foi realizado com os demais modelos considerados na Seção 3.2 do capítulo anterior. Os EQMs obtidos com cada modelo são apresentados na Tabela 6.

Tabela 6: Avaliação dos modelos no conjunto de teste

Modelo	loss	EAM	EQM
Modelo A	0,5060	0,3202	0,5060
Modelo B	0,4673	0,2653	0,4673
Modelo C	0,4913	0,2774	0,4913
Modelo Final	0,4997	0,2710	0,4500

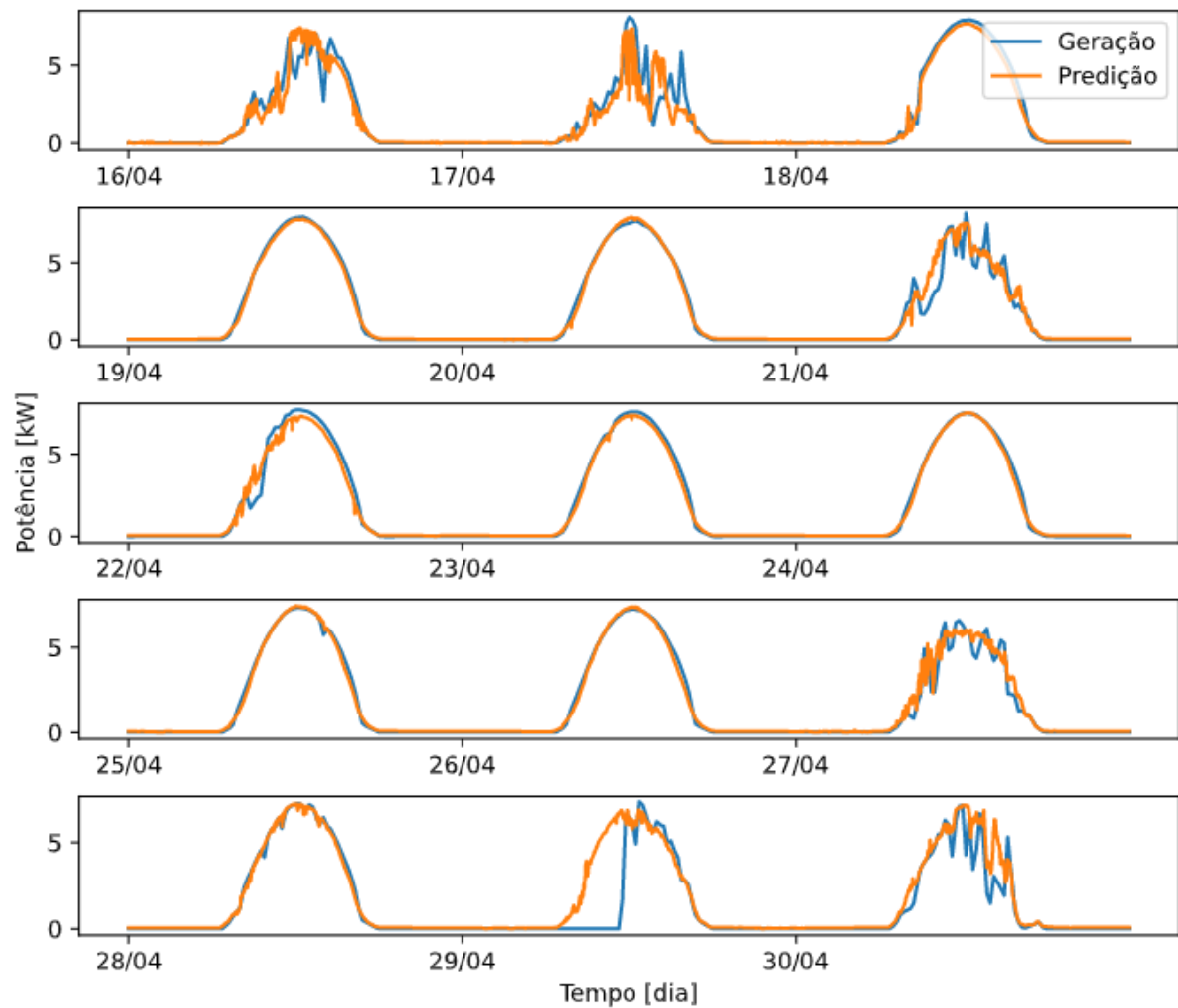
Fonte: Elaborada pelo autor.

A fim de melhor ilustrar a performance do modelo final, foi plotado um gráfico com a potência fotovoltaica medida de 16 a 30 de abril de 2018 – período correspondente ao conjunto de teste – sobreposta à predição realizada pelo modelo (Figura 35). Além disto, foi feita uma tabela comparando-se as energias prevista e efetivamente gerada em cada dia do período considerado, bem como o erro percentual (Tabela 7). O erro percentual foi calculado segundo a fórmula $\frac{\text{geração} - \text{predição}}{\text{geração}} \times 100\%$.

Algumas observações com relação ao gráfico da Figura 35: é possível notar que, em alguns trechos, as curvas de predição e de geração aparecem recortadas. Isto ocorre em dias nebulosos, devido a encobrimentos da radiação solar direta por nuvens. Contudo, embora a planta fotovoltaica e a estação meteorológica – responsável por coletar os dados ambientais – estejam no mesmo Município, elas não estão no mesmo local. Assim, elas não são igualmente afetadas pela nebulosidade. Como a predição se baseia unicamente nos dados meteorológicos, é esperado haver alguma divergência entre a curva da predição e a da geração em dias nebulosos. Todavia, em dias ensolarados, a divergência é bem menor, como se pode notar na Figura 35.

Outra anomalia ocorre no dia 29. Nele, a curva de geração sobe de zero a um valor elevado subitamente, como uma função degrau, e isto no meio do dia, quando a potência gerada já deveria estar bem acima de zero. Causas prováveis desta anomalia podem ser: falhas no sensor, perda de conexão com o servidor do portal SUNNY, etc. O modelo de aprendizado profundo, porém, alheio a quaisquer falhas na geração, previu uma curva consonante com os dados ambientais.

Figura 35: Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e previsão (laranja) do modelo final



Fonte: Elaborada pelo autor.

Tabela 7: Comparação entre energia prevista e gerada em kWh por dia e erro percentual relativos ao modelo final

Data	Predição	Geração	Erro %
16/04	37,119	37,577	1,21 %
17/04	26,507	30,680	13,60 %
18/04	48,084	50,897	5,52 %
19/04	51,222	52,362	2,17 %
20/04	50,962	50,722	-0,47 %
21/04	42,950	39,258	-9,40 %
22/04	46,711	47,877	2,43 %
23/04	47,582	49,333	3,54 %
24/04	48,348	49,018	1,36 %
25/04	47,572	47,604	0,06 %
26/04	46,733	47,245	1,08 %
27/04	38,865	35,768	-8,65 %
28/04	45,809	45,546	-0,57 %
29/04	42,945	26,702	-60,83 %
30/04	40,356	33,038	-22,15 %

Fonte: Elaborada pelo autor.

A título de comparação, um experimento semelhante foi realizado pelo autor deste trabalho, porém usando um modelo de aprendizado de máquina tradicional ao invés de aprendizado profundo [26]. Naquele experimento, foi implementado um modelo utilizando-se o método dos mínimos quadrados ordinários (*ordinary least squares*) com gradiente descendente estocástico como otimizador e regularização ℓ_2 como penalidade. Neste trabalho, aquele modelo (aqui chamado de Modelo D) é reimplementado a fim de se comparar os desempenhos do aprendizado profundo e tradicional. O Modelo D foi implementado utilizando-se o método SGDR regressor do scikit learn [27], com o quadrados do erro residual como função de perda, regularização ℓ_2 , e demais parâmetros setados para *default*. O modelo foi treinado com os conjuntos de treinamento e avaliação combinados e avaliado no conjunto de teste, resultando em um EAM de 0,4373 e um EQM de 0,5892 (não foi possível obter a métrica loss com o SGDR regressor). Estes resultados são incluídos na Tabela 8. Além disso, foi plotado um gráfico das curvas geração e predição do Modelo D (Figura 36) e feita uma comparação entre as energias gerada e prevista (Tabela 9), semelhantes às que foram feitas para o modelo final. A comparação entre os dois modelos demonstra a superioridade do aprendizado profundo sobre o aprendizado de máquina

tradicional.

Tabela 8: Avaliação dos modelos no conjunto de teste, incluindo Modelo D

Modelo	loss	EAM	EQM
Modelo A	0,5060	0,3202	0,5060
Modelo B	0,4673	0,2653	0,4673
Modelo C	0,4913	0,2774	0,4913
Modelo D		0,4373	0,5892
Modelo Final	0,4997	0,2710	0,4500

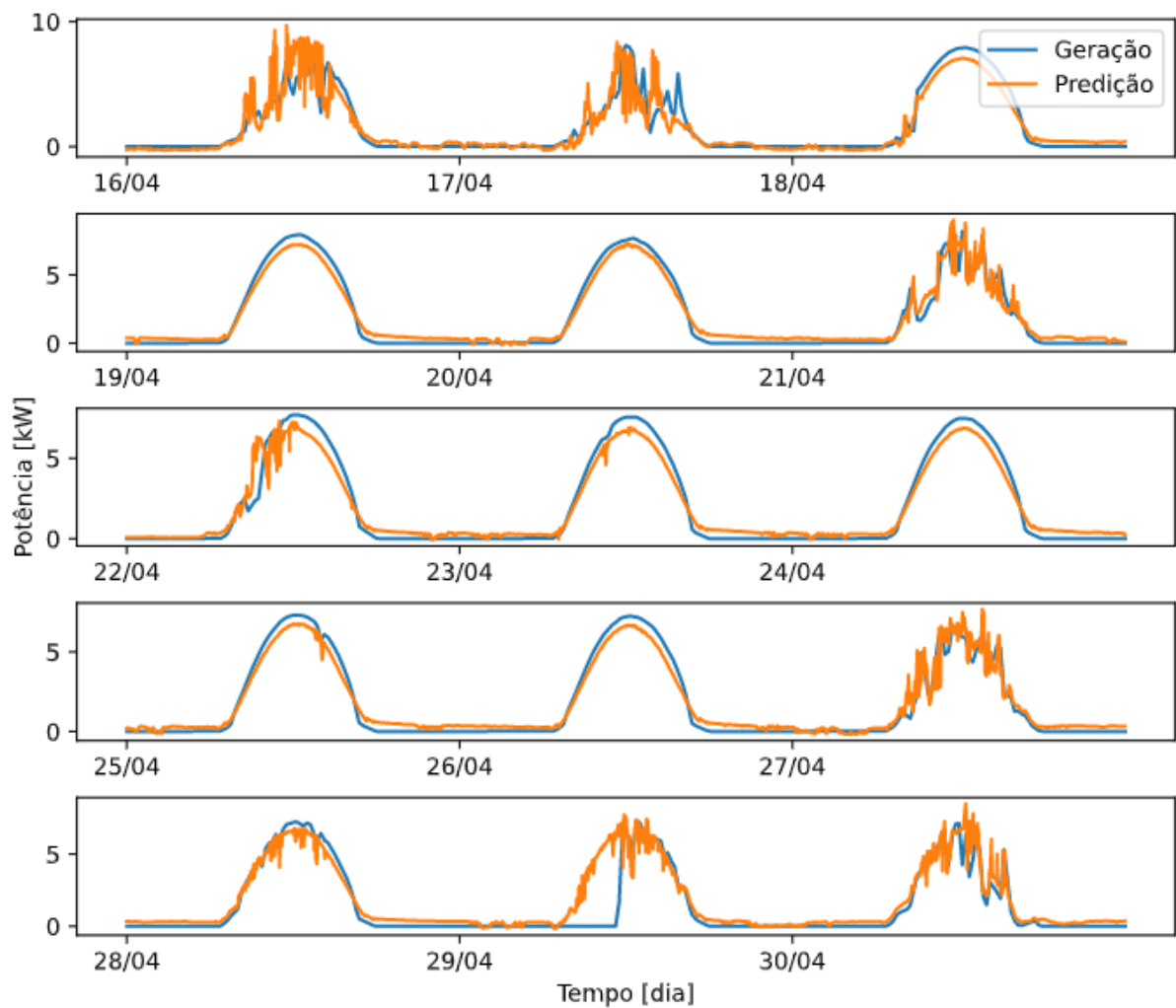
Fonte: Elaborada pelo autor.

Tabela 9: Comparação entre energia prevista e gerada em kWh por dia e erro percentual relativos ao Modelo D

Data	Predição	Geração	Erro %
16/04	40,741	37,577	-8,42 %
17/04	28,536	30,680	6,98 %
18/04	46,109	50,897	9,40 %
19/04	51,061	52,362	2,48 %
20/04	50,005	50,722	1,41 %
21/04	43,386	39,258	-10,51 %
22/04	47,786	47,877	0,19 %
23/04	47,441	49,333	3,83 %
24/04	47,957	49,018	2,16 %
25/04	46,940	47,604	1,39 %
26/04	46,184	47,245	2,24 %
27/04	39,885	35,768	-11,51 %
28/04	46,931	45,546	-3,04 %
29/04	44,376	26,702	-66,18 %
30/04	40,009	33,038	-21,09 %

Fonte: Elaborada pelo autor.

Figura 36: Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e predição (laranja) do Modelo D



Fonte: Elaborada pelo autor.

4.2 PREVISÃO SIMPLES

Os experimentos realizados até aqui foram previsões síncronas, isto é, em tempo real. Este tipo de previsão pode ser útil em casos em que se tem acesso aos dados ambientais em tempo real, mas não aos dados de geração. Assim, seria possível se fazer uma estimativa da geração de um sistema fotovoltaico em tempo real sem se ter acesso ao mesmo. Contudo, seria interessante para o operador do sistema de energia elétrica obter uma previsão assíncrona, com alguma antecedência, da planta considerada. Os experimentos a seguir são previsões com 30 minutos e 1 hora de antecedência. Foram utilizados o mesmo modelo final e conjunto de dados, porém com uma defasagem entre os dados de entrada e saída de 30 minutos e 1 hora, respectivamente.

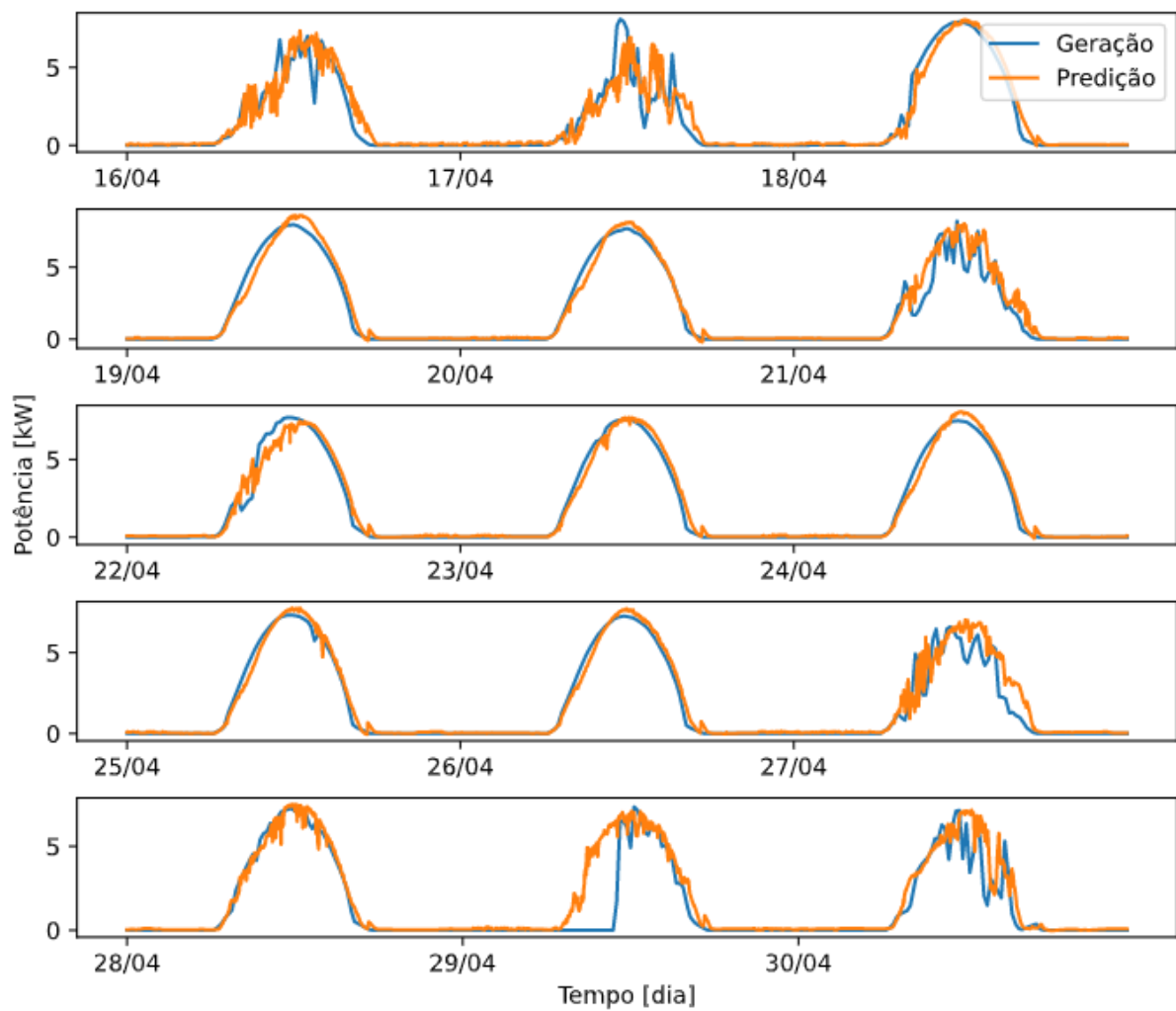
A Tabela 10 apresenta os desempenhos destes experimentos nas métricas loss, EAM e EQM avaliadas no conjunto de teste. As Tabelas 11 e 12 mostram as comparações entre as energias previstas e geradas e os erros percentuais para cada caso, e as Figuras 37 e 38 mostram as curvas de geração e previsão dos mesmos.

Tabela 10: Avaliação do modelo final no conjunto de teste com 30 minutos e 1 hora de antecedência

Tempo de antecedência	loss	EAM	EQM
30 minutos	0,6486	0,3691	0,5731
1 hora	1,1128	0,6011	0,9979

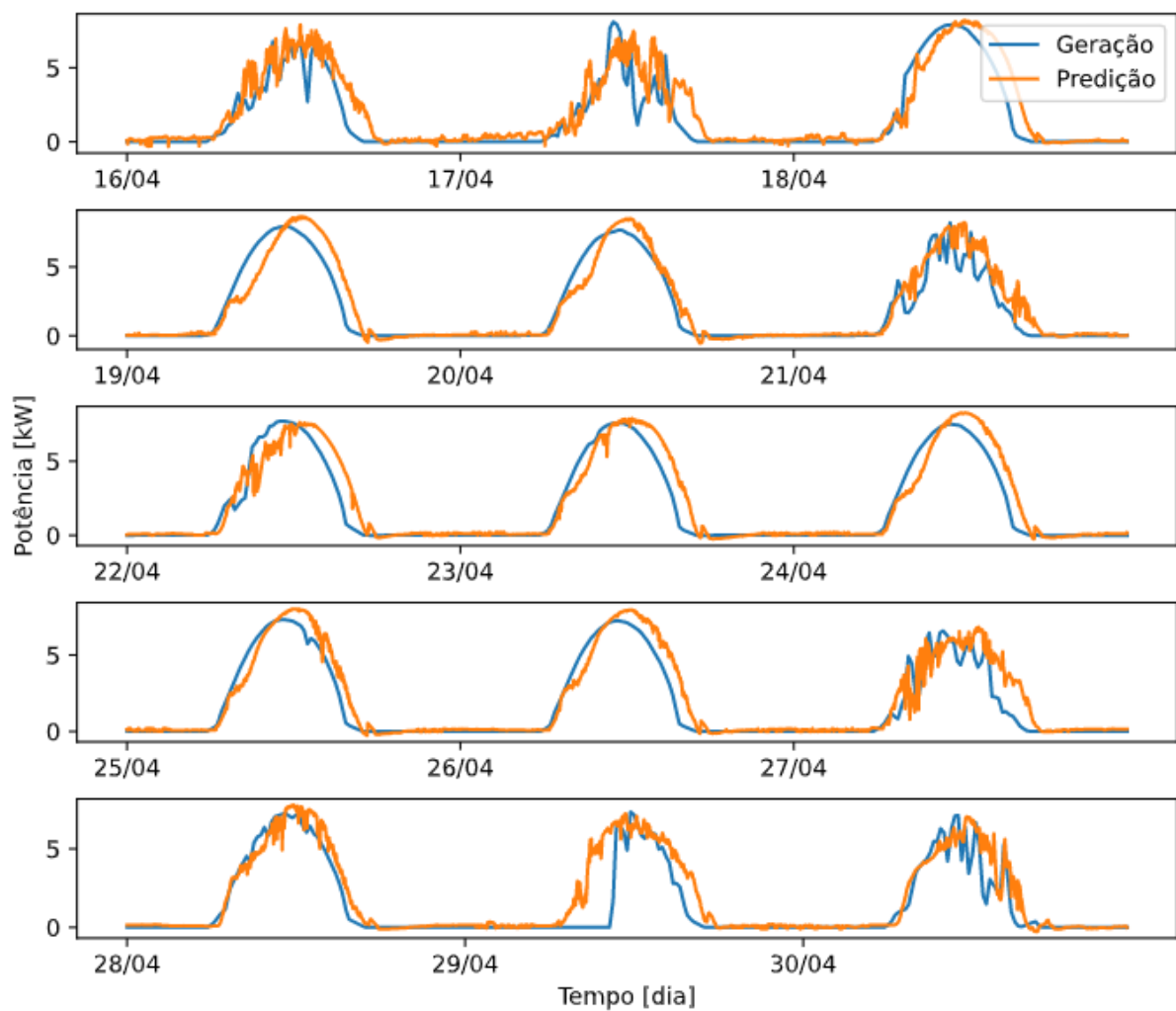
Fonte: Elaborada pelo autor.

Figura 37: Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e previsão (laranja) do modelo final com 30 minutos de antecedência



Fonte: Elaborada pelo autor.

Figura 38: Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e previsão (laranja) do modelo final com 1 hora de antecedência



Fonte: Elaborada pelo autor.

Tabela 11: Comparação entre energia prevista e gerada em kWh por dia e erro percentual relativos ao modelo final, previsão com 30 minutos de antecedência

Data	Predição	Geração	Erro %
16/04	42,040	37,577	-11,87 %
17/04	32,560	30,680	-6,12 %
18/04	51,841	50,897	-1,85 %
19/04	53,947	52,362	-3,02 %
20/04	51,105	50,722	-0,75 %
21/04	48,381	39,258	-23,23 %
22/04	48,789	47,877	-1,90 %
23/04	50,218	49,333	-1,79 %
24/04	50,196	49,018	-2,40 %
25/04	48,492	47,604	-1,86 %
26/04	48,873	47,245	-3,44 %
27/04	44,454	35,768	-24,28 %
28/04	46,749	45,546	-2,64 %
29/04	45,368	26,702	-69,89 %
30/04	39,977	33,038	-20,99 %

Fonte: Elaborada pelo autor.

Tabela 12: Comparação entre energia prevista e gerada em kWh por dia e erro percentual relativos ao modelo final, previsão com 1 hora de antecedência

Data	Predição	Geração	Erro %
16/04	51,161	37,577	-36,15 %
17/04	44,041	30,680	-43,55 %
18/04	57,255	50,897	-12,49 %
19/04	54,843	52,362	-4,73 %
20/04	51,103	50,722	-0,75 %
21/04	51,592	39,258	-31,41 %
22/04	52,908	47,877	-10,50 %
23/04	54,656	49,333	-10,33 %
24/04	54,085	49,018	-8,94 %
25/04	51,860	47,604	-14,24 %
26/04	53,977	47,245	-28,38 %
27/04	45,921	35,768	-9,42 %
28/04	49,837	45,546	-2,64 %
29/04	47,414	26,702	-77,56 %
30/04	39,053	33,038	-18,20 %

Fonte: Elaborada pelo autor.

4.3 PREVISÃO COM O CONJUNTO DE DADOS REESTRUTURADO

Com o intuito de melhorar a performance da previsão, foi tentada uma abordagem diferente: reestruturar o conjunto de dados. Lembrando que o conjunto de dados consiste em seis atributos (dados ambientais) e uma saída (potência), foi adicionado um novo atributo: a hora do dia, haja vista que esta é uma boa preditora da geração fotovoltaica. Além da hora do dia, foram utilizados dados ambientais com defasagem de 1T, 2T, 3T, 4T e 5T – sendo T o período de antecedência da previsão –, totalizando $1+5 \times 6 = 31$ atributos. Note-se que a hora do dia ainda está sincronizada com a potência, porém os dados ambientais são de 1, 2, 3, 4 e 5 T antes da leitura da potência. Desta forma, se $T = 1$ hora, a previsão será com 1 hora de antecedência, e assim por diante. Devido ao aumento do número de atributos, houve um aumento do número de parâmetros treináveis de 1313 para 2113 no modelo final. Novamente, realizaram-se previsões com 30 minutos e 1 hora de antecedência. Os resultados são mostrados a seguir.

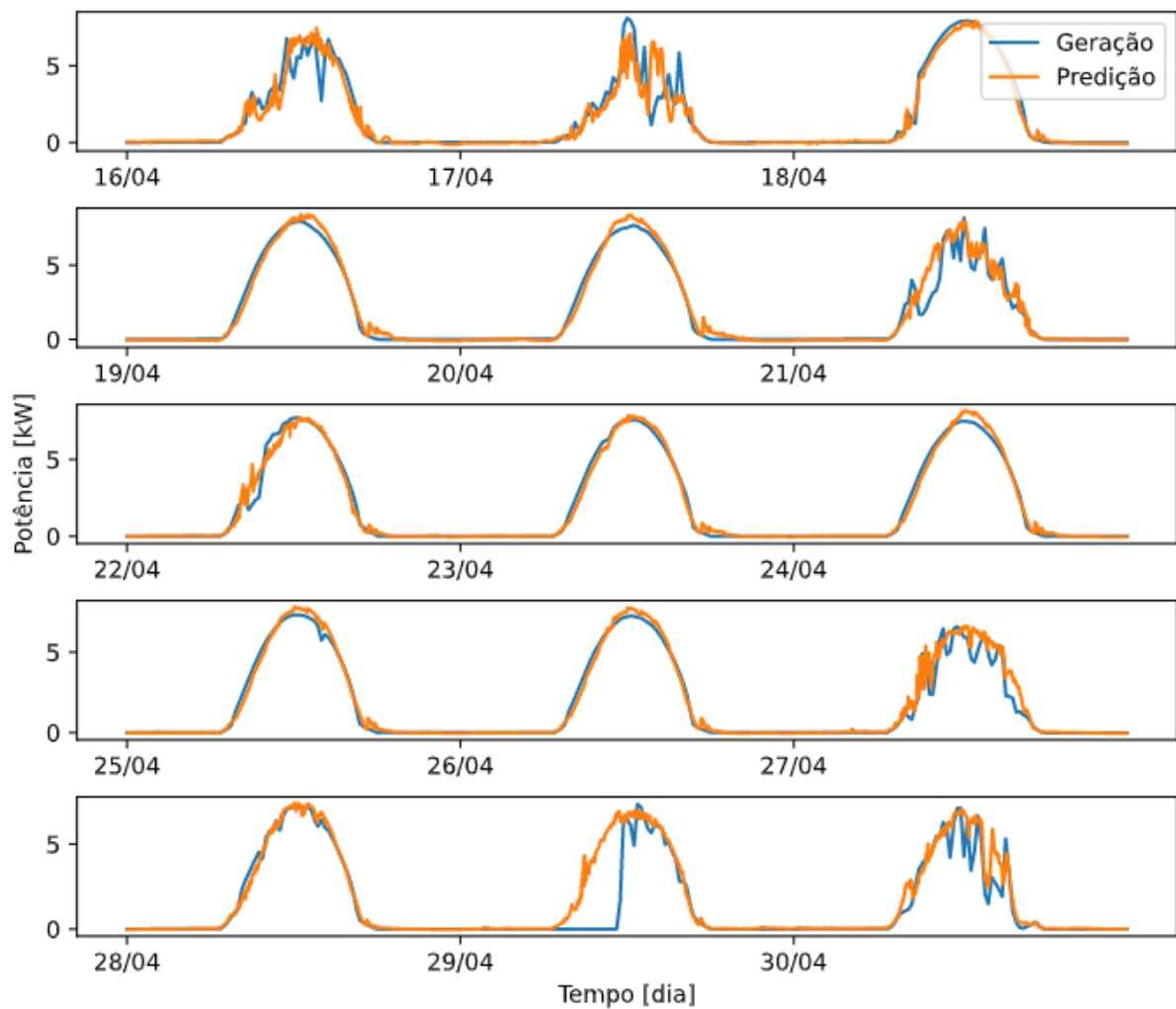
A Tabela 13 apresenta as métricas loss, EAM e EQM avaliadas no conjunto de teste. As Tabelas 14 e 15 mostram as comparações entre as energias previstas e geradas e as Figuras 39 e 40 mostram as curvas de geração e predição.

Tabela 13: Avaliação do modelo final no conjunto de teste com 30 minutos e 1 hora de antecedência com o conjunto de dados reestruturado

Tempo de antecedência	loss	EAM	EQM
30 minutos	0,5086	0,2815	0,4416
1 hora	0,5346	0,2965	0,4741

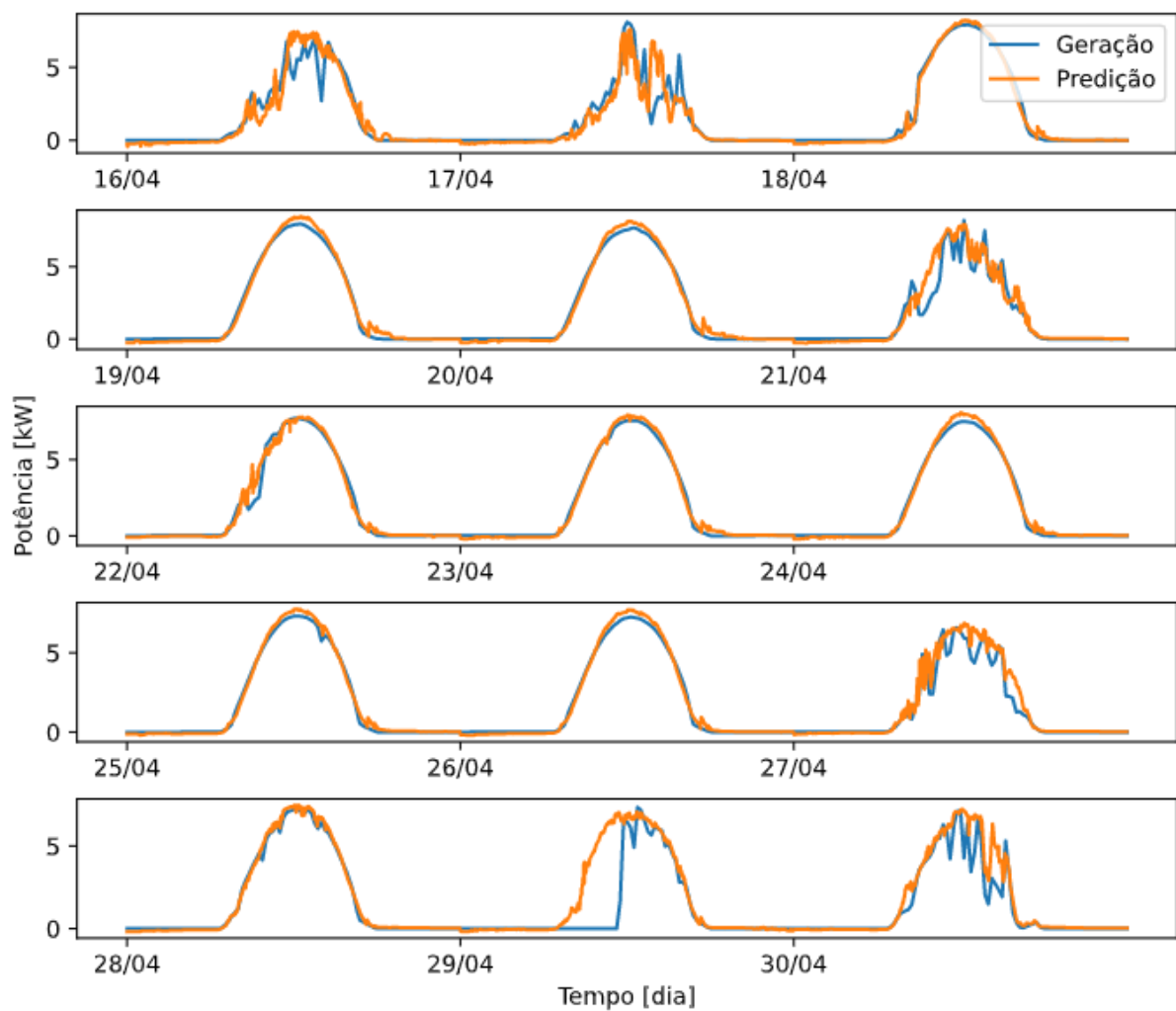
Fonte: Elaborada pelo autor.

Figura 39: Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e previsão (laranja) do modelo final com 30 minutos de antecedência com o conjunto de dados reestruturado



Fonte: Elaborada pelo autor.

Figura 40: Curva de geração (azul) do Sistema “Sede” de 16 a 30 de abril de 2018 e previsão (laranja) do modelo final com 1 hora de antecedência com o conjunto de dados reestruturado



Fonte: Elaborada pelo autor.

Tabela 14: Comparação entre energia prevista e gerada em kWh por dia e erro percentual relativos ao modelo final com conjunto de dados reestruturado, previsão com 30 minutos de antecedência

Data	Predição	Geração	Erro %
16/04	37,909	37,577	-0,88 %
17/04	28,670	30,680	6,54 %
18/04	48,499	50,897	4,71 %
19/04	52,908	52,362	-1,04 %
20/04	52,375	50,722	-3,25 %
21/04	43,192	39,258	-10,01 %
22/04	47,198	47,877	1,41 %
23/04	48,543	49,333	1,59 %
24/04	49,480	49,018	-0,94 %
25/04	48,414	47,604	-1,70 %
26/04	47,341	47,245	-0,20 %
27/04	41,659	35,768	-16,46 %
28/04	45,499	45,546	0,10 %
29/04	44,077	26,702	-65,06 %
30/04	38,595	33,038	-16,81 %

Fonte: Elaborada pelo autor.

Tabela 15: Comparação entre energia prevista e gerada em kWh por dia e erro percentual relativos ao modelo final com conjunto de dados reestruturado, previsão com 1 hora de antecedência

Data	Predição	Geração	Erro %
16/04	38,089	37,577	-1,36 %
17/04	27,613	30,680	9,99 %
18/04	49,747	50,897	2,25 %
19/04	53,368	52,362	-1,92 %
20/04	51,937	50,722	-2,39 %
21/04	42,782	39,258	-8,97 %
22/04	48,805	47,877	-1,93 %
23/04	50,179	49,333	-1,71 %
24/04	50,254	49,018	-2,52 %
25/04	48,998	47,604	-2,92 %
26/04	48,322	47,245	-2,28 %
27/04	41,726	35,768	-16,65 %
28/04	46,637	45,546	-2,39 %
29/04	44,577	26,702	-66,93 %
30/04	40,412	33,038	-22,31 %

Fonte: Elaborada pelo autor.

5 CONSIDERAÇÕES FINAIS

Neste trabalho, a potência gerada por um sistema fotovoltaico é prevista utilizando-se aprendizado de máquina profundo. Inicialmente, apresenta-se um breve resumo sobre energia solar e sistemas fotovoltaicos. Em seguida, apresentam-se os principais conceitos sobre aprendizado de máquina. Na sequência, são descritos o conjunto de dados e suas peculiaridades e é relatada a implementação da rede neural. Os dados utilizados no treinamento da rede, adquiridos de fontes distintas, foram submetidos a pré-processamento e feature engineering. Uma série de procedimentos foi seguida para se garantir que a rede não apresentasse overfitting. Partindo-se de um modelo básico, estabeleceu-se um número máximo de épocas de treinamento, monitorando-se a perda no conjunto de validação. Depois, foi incrementado o número de camadas e unidades de camada progressivamente, até a performance no conjunto de validação degradar. Finalmente, foram adicionadas regularização ℓ_2 e dropout. O modelo final foi treinado com o equivalente a 46 dias de dados de treinamento e avaliado em 15. Com este modelo, foi possível obter uma previsão em tempo real da potência gerada por uma planta fotovoltaica com um EQM de 0,45. Posteriormente, o conjunto de dados foi reestruturado a fim de se obter previsões com antecedência. Para tal, o conjunto de dados foi ampliado com um novo atributo, a hora do dia, e foram usados os dados ambientais de 1, 2, 3, 4 e 5 T atrás – sendo T o período de antecedência da previsão –, aumentando-se o número de atributos para 31. Com este conjunto de dados reestruturado, foram feitas previsões com 30 minutos e 1 hora de antecedência com EQMs de 0,4416 e 0,4741, respectivamente.

5.1 PROPOSTAS PARA TRABALHOS FUTUROS

A rede neural utilizada neste trabalho é do tipo feedforward, que, segundo Goodfellow *et al.* [12], é um dos tipos mais comuns. Na revisão do estado da arte (Seção 2.5), porém, constatou-se a existência de outras abordagens de aprendizado de máquina bastante interessantes (por exemplo, rede neural evolutiva [17]). Para trabalhos futuros, propõe-se a utilização de abordagens mais em paridade com o estado da arte.

Outra proposta para um trabalho futuro seria aumentar o conjunto de treinamento. Neste trabalho foram utilizados apenas 46 dias de dados de treinamento, porém, o conjunto de dados reunido totaliza 16 meses. Estima-se que, com um conjunto

de treinamento maior, seria possível obter resultados ainda melhores.

Ainda mais uma proposta seria a realização de experimentos com alguns dos atributos que foram removidos por feature engineering. Em particular, os atributos relacionados ao vento, uma vez que estes podem ser preditores de chuva e, conseqüentemente, podem ser preditores da geração fotovoltaica em um prazo de tempo maior.

REFERÊNCIAS

- [1] SU, D.; BATZELIS, E.; PAL, B. Machine learning algorithms in forecasting of photovoltaic power generation. *2019 International Conference on Smart Energy Systems and Technologies (SEST)*, Porto, Portugal, p. 1–6, 2019.
- [2] TEO, T. T. et al. Forecasting of photovoltaic power using regularized ensemble extreme learning machine. *2016 IEEE Region 10 Conference (TENCON)*, Singapore, p. 455–458, 2016.
- [3] ANEEL. *Resolução Normativa Nº 733, de 06 de setembro de 2016*.
- [4] ANEEL. *Resolução Normativa Nº 484, de 17 de abril de 2012*.
- [5] CHOLLET, F. *Deep Learning with Python*. Shelter Island, Nova York, EUA: Manning Publications, 2018.
- [6] PINHO, J. T.; GALDINO, M. A. *Manual de Engenharia para Sistemas Fotovoltaicos: Edição Revisada e Atualizada*. Rio de Janeiro: CEPEL – CRESESB, 2014.
- [7] GLOBAL SOLAR ATLAS. *solarworld*. Disponível em: <https://globalsolaratlas.info/download/world>. Acesso em: 6 nov. 2020.
- [8] GLOBAL SOLAR ATLAS. *solarbr*. Disponível em: <https://globalsolaratlas.info/download/brazil>. Acesso em: 6 nov. 2020.
- [9] COURSERA. *Solar Energy Basics*. Disponível em: <https://www.coursera.org/learn/solar-energy-basics/lecture/boJYt/adjusting-for-location-and-system-losses>. Acesso em: 13 nov. 2020.
- [10] WATT, J.; BORHANI, R.; KATSAGGELOS, A. K. *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge, Cambridgeshire, Reino Unido: Cambridge University Press, 2016.
- [11] MURPHY, K. P. *Machine Learning: A Probabilistic Perspective*. Cambridge, Massachusetts, EUA: The MIT Press, 2012.
- [12] GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. Cambridge, Massachusetts, EUA: The MIT Press, 2016. <http://www.deeplearningbook.org>.
- [13] ABU-MOSTAFA, Y.; MAGDON-ISMAIL, M.; LIN, H.-T. *Learning from Data: a Short Course*. [S.l.]: AMLbook, 2020. [AMLbook.com](http://amlbook.com).
- [14] COURSERA. *Neural Networks and Deep Learning*. Disponível em: <https://www.coursera.org/learn/neural-networks-deep-learning>. Acesso em: 13 nov. 2020.
- [15] TECHNICAL FRIDAYS. *Don't use sigmoid: Neural Nets*. Disponível em: <https://kharshit.github.io/blog/>. Acesso em: 17 fev. 2021.

- [16] KERAS. *RMSPProp*. Disponível em: <https://keras.io/api/optimizers/rmsprop/>. Acesso em: 16 fev. 2021.
- [17] OGAWA, S.; MORI, H. Application of evolutionary deep neural network to photovoltaic generation forecasting. *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*, Sapporo, Japan, p. 1–4, 2019.
- [18] KAFFASH, M.; DECONINCK, G. Ensemble machine learning forecaster for day ahead pv system generation. *2019 IEEE 7th International Conference on Smart Energy Grid Engineering (SEGE)*, Oshawa, ON, Canada, p. 92–96, 2019.
- [19] THEOCHARIDES, S. et al. Machine learning algorithms for photovoltaic system power output prediction. *2018 IEEE International Energy Conference (ENERGYCON)*, Limassol, Cyprus, p. 1–6, 2018.
- [20] JIANG, Y. et al. Research on predicting the short-term output of photovoltaic (pv) based on extreme learning machine model and improved similar day. *2019 IEEE Innovative Smart Grid Technologies - Asia (ISGT Asia)*, Chengdu, China, p. 3691–3695, 2019.
- [21] INPE. *Sistema de Organização dos Dados Ambientais*. Disponível em: <http://sonda.ccst.inpe.br/>. Acesso em: 14 ago. 2020.
- [22] SMA. *SUNNY Portal*. Disponível em: <https://www.sunnyportal.com/Templates/Start.aspx?ReturnUrl=%2f>. Acesso em: 14 ago. 2020.
- [23] CICLO VIVO. *Empresa em Florianópolis deixará de emitir 200 toneladas de CO2 com painéis solares*. Disponível em: <https://ciclovivo.com.br/planeta/desenvolvimento/florianopolis-deixara-de-emitir-200-toneladas-de-co2-com-usina-solar/>. Acesso em: 14 ago. 2020.
- [24] HORA DE VERÃO. *Decreto Nº 733, de 08 de setembro de 2008*.
- [25] KERAS. *Keras Tuner*. Disponível em: <https://keras-team.github.io/keras-tuner/documentation/tuners/>. Acesso em: 16 fev. 2021.
- [26] SEHN, S. Relatório de estágio longo: Predição de geração fotovoltaica com aprendizado de máquina. UFSC, Florianópolis, 2019.
- [27] SCIKIT-LEARN. *SGDRegressor*. Disponível em: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.SGDRegressor.html#sklearn.linear_model.SGDRegressor. Acesso em: 29 set. 2020.