

Redes LSTM aplicadas a previsão de energia solar

André F. C.¹, Marcelo H. I.¹, Rodrigo S. H.¹, Victor G. O. M. N.¹

¹Escola de Artes, Ciências e Humanidades – Universidade de São Paulo (USP)

Abstract. *The purpose of this article is to present a study on the application of Long Short-Term Memory (LSTM) neural networks on the task of predicting solar energy using measured hourly variables obtained from stations in the southeastern region of Brazil. The focus of the tests was on short-term (nowcasting) forecasts, ranging up to one day. The objective of this study is to verify the viability of the application of this neural network and to bring insights to possible future studies.*

Resumo. *O presente artigo visa trazer um estudo sobre a aplicação de redes neurais do tipo Long Short-Term Memory (LSTM) para a tarefa de previsão de energia solar utilizando variáveis medidas de hora em hora, obtidas em estações da região sudeste do Brasil. O foco dos testes foi em previsões de curto prazo (nowcasting), variando até um dia. O objetivo desse estudo é verificar a viabilidade da aplicação dessa rede e trazer insights para possíveis estudos futuros.*

1. INTRODUÇÃO

Segundo o *World Energy Assessment: Energy and the Challenge of Sustainability* [Program and Council 2000], o potencial mínimo de energia solar anual é de 1575 exajoules no mundo e a taxa de consumo energético mundial projetado para 2020 está entre 590 e 1050 exajoules. A quantidade captável de energia depende das tecnologias aplicadas na conversão, mas ainda assim, há indícios de que a energia solar, além de ser considerada limpa e renovável, tem potencial para atender a demanda de energia mundial sem a necessidade de outras fontes.

Tendo em vista esse potencial, o presente trabalho pretende contribuir para o desenvolvimento e gerenciamento de tecnologias na captação de energia solar, buscando demonstrar a aplicabilidade de uma Rede Neural Artificial do tipo *Long Short-Term Memory* (LSTM) na tarefa de prever a quantidade de energia solar com base em outros atributos climáticos.

Vale mencionar que a escolha do tema foi inspirada no *Google Sunroof Project* [Google 2015] e a escolha da LSTM deu-se por conta do potencial demonstrado em previsões de outros tipos de séries temporais.

2. METODOLOGIA

Nesta seção são apresentados o conjunto de dados e as etapas do pré-processamento aplicado, o algoritmo LSTM para criar o modelo preditivo, bem como o código utilizado em python e a motivação da escolha dos atributos e parâmetros de treino.

2.1. Conjunto de dados

Para a confecção deste trabalho, foi utilizado o *dataset Hourly Weather Surface - Brazil (Southeast region)*[Araújo et al. 2018]. Os dados são compostos de variáveis climáticas que foram coletadas de hora em hora por 122 estações meteorológicas localizadas em diferentes locais do sudeste brasileiro. O período de observações varia de estação para estação, havendo dados de 2006 a 2016.

2.1.1. Pré-processamento

O conjunto de dados original continha todas as observações de todas as estações em uma mesma planilha. Portanto, o primeiro passo foi pré-selecionar, aleatoriamente, 12 estações, dividindo-as em arquivos diferentes e separando as medidas de cada estação por ano.

Estações Pré-selecionadas

- 178 - São Gonçalo
- 306 - Alfredo Chaves
- 321 - Ouro Branco
- 335 - Teófilo Otoni
- 347 - Porteirinha
- 375 - Arraial do Cabo
- 376 - Campos dos Goytacazes
- 381 - Teresópolis
- 390 - Silva Jardim
- 392 - São Paulo
- 412 - Ariranhã
- 413 - Ibitinga

2.1.2. Tratando *missing data*

O maior problema descoberto no conjunto de dados foi que em todas as estações, não havia medidas de energia solar nos períodos noturnos, pois aparentemente os aparelhos eram desligados. Além disso, algumas estações (em alguns anos) passavam dias sem nenhuma medida, possivelmente por algum defeito nos medidores ou outro tipo de contratempo. Assim, o segundo passo no pré-processamento foi o tratamento do excesso de dados faltantes.

Observou-se que o período em que faltavam medidas variava de estação para estação e de tempo em tempo, por isso, a porcentagem de zeros para cada ano e para cada estação foi analisada, conforme tabela abaixo. Com isso, decidiu-se que a estação de São Paulo seria um bom ponto de partida para a etapa exploratória do algoritmo.

Ao pesquisar algumas formas de tratamento desse tipo de *missing data* na literatura [Lipton et al. 2016b][Lipton et al. 2016a], a estratégia de remoção de um período fixo de medidas foi escolhido. Ou seja, para todas as estações e anos analisados, foram removidas as medidas realizadas entre 23:00 e 8:00. No entanto, como o período variava, sobraram alguns resquícios de zeros.

Estações escolhidas e a contagem de zeros		2006	2007	2008	2009	2010	2011	2012	2013	2014	2015	2016	Total
Alfredo	Contagem	0	4117	4087	3953	4008	4020	4102	4092	4305	4094	0	36778
	Média	0.00%	47.00%	46.53%	45.13%	45.76%	45.90%	46.70%	46.72%	49.15%	46.74%	0.00%	46.62%
Aririnha	Contagem	0	0	0	0	0	0	4246	4336	3692	8581	0	20855
	Média	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	48.34%	49.50%	42.15%	97.97%	0.00%	59.48%
Arraial	Contagem	0	4164	4205	4299	4114	4132	5046	4129	4089	4140	0	38318
	Média	0.00%	47.53%	47.88%	49.08%	46.97%	47.17%	57.45%	47.14%	46.68%	47.27%	0.00%	48.57%
Campos	Contagem	1073	4063	3996	4151	4424	4395	4117	4346	4106	4122	3150	41943
	Média	45.62%	46.39%	45.50%	47.39%	50.51%	50.18%	46.87%	49.62%	46.88%	47.06%	47.91%	47.76%
Ibitinga	Contagem	0	0	4141	5221	5149	4117	4127	4512	4670	4104	0	36041
	Média	0.00%	0.00%	47.14%	59.61%	58.79%	47.00%	46.99%	51.51%	53.32%	46.85%	0.00%	51.39%
Ouro Branco	Contagem	1760	4129	3968	4256	4107	4117	4114	4119	4107	4100	3161	41938
	Média	46.71%	47.14%	45.18%	48.59%	46.89%	47.00%	46.84%	47.03%	46.89%	46.81%	48.08%	47.00%
Porteirinha	Contagem	0	574	4158	4132	4157	4143	4144	5213	4664	4266	3063	38514
	Média	0.00%	46.90%	47.34%	47.17%	47.46%	47.30%	47.18%	59.52%	53.25%	48.70%	46.59%	49.42%
Sao Goncalo	Contagem	0	0	4124	7215	4049	5179	4030	4085	4016	4020	0	36718
	Média	0.00%	0.00%	46.95%	82.37%	46.23%	59.13%	45.88%	46.64%	45.85%	45.90%	0.00%	52.36%
SaoPaulo	Contagem	0	4133	4200	4102	4115	4167	4137	4116	4215	4099	0	37284
	Média	0.00%	47.18%	47.82%	46.83%	46.98%	47.57%	47.10%	46.99%	48.12%	46.80%	0.00%	47.26%
Silva Jardim	Contagem	0	0	0	0	0	0	0	0	0	1421	3124	4545
	Média	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	46.99%	47.51%	47.34%
Teófilo Otoni	Contagem	0	4142	4121	4121	4113	4245	4133	4100	4130	5092	0	38197
	Média	0.00%	47.28%	46.92%	47.05%	46.96%	48.46%	47.06%	46.81%	47.15%	58.13%	0.00%	48.42%
Teresópolis	Contagem	654	4123	4044	4158	4138	4378	4158	4175	4157	4173	3399	41557
	Média	44.67%	47.07%	46.04%	47.47%	47.24%	49.98%	47.34%	47.67%	47.46%	47.64%	51.70%	47.81%

Figura 1. Quantidade de zeros de cada estação

Com base na nova proporção de zeros e após a etapa exploratória da LSTM, o ano de 2012 foi escolhido por tratar-se de um ano presente em quase todas as estações (menos Silva Jardim) e por ser um ano bissexto.

Além disso, dentre as 12 estações, realizou-se um novo recorte de 4 estações que, para o ano de 2012, não possuíam intervalos sem medidas e possuíam número reduzido de zeros, pois é de se esperar que essas trariam resultados mais coerentes com a realidade.

As quatro estações escolhidas são: Alfredo Chaves, Ibitinga, São Gonçalo e São Paulo. Todos os testes foram feitos nessas estações, para dados obtidos no ano de 2012.

2.1.3. Seleção de atributos

Cada uma das estações possuía 31 variáveis e, dessas, foram selecionadas 17 para preparar o modelo. Atributos identificadores como Id da estação, latitude e longitude, foram retiradas porque acreditava-se que elas não trariam conteúdo relevante para o modelo. Outros tipos de variáveis como algumas formas de registros de datas foram resumidas em apenas uma variável para facilitar a compreensão de quando era a medição efetuada.

Variáveis originais

- wsid - Id da estação climática - qualitativa nominal
- wnm - Nome da estação (usualmente nome da cidade ou apelido) - qualitativa nominal
- elvt - Elevação - quantitativa contínua
- lat - Latitude - quantitativa contínua
- lon - Longitude - quantitativa contínua
- inme - Número da estação (número do INMET) para o local - qualitativa nominal
- city - Cidade - qualitativa nominal
- prov - Estado (Província) - qualitativa nominal
- mdct - Data e hora da observação (data completa: data + hora) - qualitativa ordinal
- date - Data da observação - qualitativa ordinal

- yr - Ano (2000-2016) - qualitativa ordinal
- mo - Mês (0-12) - qualitativa ordinal
- da - Dia (0-31) - qualitativa ordinal
- hr - Hora (0-23) - qualitativa ordinal
- prcp - Quantidade de precipitação por milímetros (última hora) - quantitativa contínua
- stp - Pressão do ar na hora dada em hPa para décimos (instante) - quantitativa contínua
- smax - Pressão máxima de ar na última hora em hPa para décimos - quantitativa contínua
- smin - Pressão mínima de ar na última hora em hPa para décimos - quantitativa contínua
- gbrd - Radiação solar KJ/m^2 - quantitativa contínua
- temp - Temperatura do ar (instante) em graus Célsius - quantitativa contínua
- dewp - Temperatura do ponto de condensação (instante) em graus Célsius - quantitativa contínua
- tmax - Temperatura máxima para a última hora em graus Célsius - quantitativa contínua
- dmax - Temperatura do ponto de condensação máxima na última hora em graus Célsius - quantitativa contínua
- tmin - Temperatura mínima para a última hora em graus Célsius - quantitativa contínua
- dmin - Temperatura do ponto de condensação mínima na última hora em graus Célsius - quantitativa contínua
- hmdy - Humidade relativa em % (instante) - quantitativa discreta
- hmax - Máxima humidade relativa da temperatura para a última hora em % - quantitativa discreta
- hmin - Mínima humidade relativa da temperatura para a última hora em % - quantitativa discreta
- wdsp - Velocidade do vento em m/s - quantitativa contínua
- wdct - Direção do vento em graus (0° - 360°) - quantitativa contínua
- gust - Velocidade máxima do vento em m/s - quantitativa contínua

Como mencionado na seção 2.1.2, a estação de São Paulo foi escolhida para explorar os parâmetros do algoritmo. Além disso, os dados dessa estação foram utilizados para que as correlações entre as variáveis fossem obtidas. Sabendo que a variável a ser prevista é a energia solar, é interessante destacar a correlação dos outros 15 atributos com essa variável. Com base nos valores calculados, três combinações de atributos foram criadas, conforme figura 2.

O conjunto A é formado por todos os 17 atributos, sem nenhum recorte. Do conjunto A para o B, foram removidos atributos com menores correlações deixando 10 variáveis que possuíam uma correlação mais alta com a radiação solar e, do B para o C, os atributos com correlações parecidas - como por exemplo, umidade, hmin (umidade mínima) e hmax (umidade máxima) que possuem uma correlação redundante para a variável de radiação solar - foram removidos sobrando 5 variáveis no conjunto, como é possível observar na figura 2.

A					
B				Ponto de orvalho mínimo	-0.1658
C		hmin	-0.5893	Ponto de orvalho	-0.1214
Umidade	-0.6163	hmax	-0.5277	Ponto de orvalho Máximo	-0.0246
Direção do vento	0.2442	Velocidade máxima do vento	0.2845	Pressão média do ar	0.0482
Velocidade do vento	0.2706	Temperatura mínima	0.504	Pressão mínima do ar	0.0495
Temperatura	0.5754	Temperatura máxima	0.561	Pressão máxima do ar	0.0499
Energia Solar	1				

Figura 2. Combinação de atributos

Para cada combinação de atributos, todos os testes descritos na seção 7 foram realizados, a fim de comparar a influência dos atributos removidos no desempenho do modelo e buscar a melhor combinação.

2.2. Long Short-Term Memory (LSTM)

O problema de previsão da quantidade de energia solar, assim como outras previsões de séries temporais, pode ser dividido em três categorias: para previsão de algumas horas, *Nowcasting*, para prever alguns dias, *Short-term forecasting* e previsões de longo prazo, *Long-term forecasting*. Além disso, as formas de lidar com esse problema são abordagens baseadas em modelos físicos, estatísticos e híbridos.

Os físicos são baseados em NWP (*Numeric Weather Prediction* (simulações de cenários climáticos), observações via satélite de nuvens com reconhecimento de padrões em imagens e outros métodos que envolvem algoritmos de alta complexidade e que geralmente exigem processamento em larga escala.

Já os modelos estatísticos são subdivididos em métodos puramente estatísticos e modelos baseados em aprendizagem. Ou seja, técnicas de aproximação de funções que descrevem o comportamento de variáveis reais e outros modelos matemáticos que possam ser ajustados para descrever a realidade. São exemplos dessas técnicas: ARIMA (*Auto Regressive Integrated Moving Average*) e regressões múltiplas para métodos estatísticos e algoritmos genéticos e redes neurais artificiais como modelos baseados em aprendizagem.

Por fim, a abordagem híbrida é uma combinação de modelos físicos e estatísticos, que se complementem e tragam melhores resultados que cada um deles, isoladamente.

Para cada tipo de previsão, existem técnicas mais recomendadas na literatura, ou mais usadas. Por exemplo, para previsões de longo prazo, é comum ser utilizada a abordagem baseada em modelos físicos ou híbrida. Tendo em vista que o poder computacional exigido por essa abordagem e o tempo demandado para implementação de testes, optou-se por realizar previsões de algumas horas, até um dia.

Sabendo que dentre as técnicas mais utilizadas para abordar o problema de *Now-*

casting estão as redes neurais artificiais, foi escolhido um tipo especial de rede que, diferente das redes *multilayer perceptron* (MLP) tradicionais, é capaz de armazenar dependências de longo termo, a *Long Short-Term Memory* (LSTM).

Uma rede neural do tipo *multilayer perceptron* é dividida em camadas, uma sensorial que recebe os atributos do dado de entrada; uma ou mais camadas escondidas, que aplicam pesos e funções nos atributos para aproximá-los da saída desejada. Por fim, uma camada de saída recebe os resultados após o processamento das camadas escondidas e também realiza uma série de operações que trazem o resultado final, que pode ser uma classificação ou uma previsão, por exemplo.

Além disso, cada camada é composta por neurônios, uma abstração inspirada na biologia. Na prática porém, as camadas podem ser vetores que guardam conjuntos de valores, os quais são utilizados no processamento das saídas.

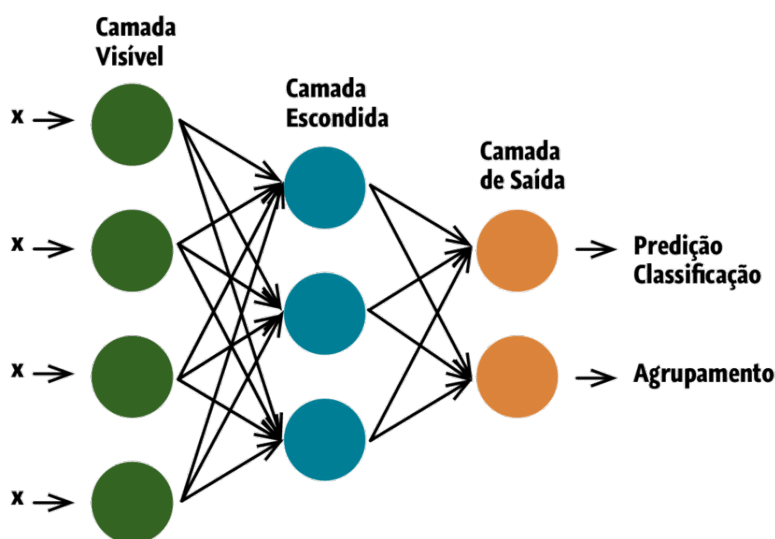


Figura 3. Multilayer Perceptron (MLP)

Durante o treinamento de uma rede neural, é necessário determinar uma quantidade de épocas em que os dados serão lidos pelo algoritmo, que são o número de iterações da previsão sobre o conjunto de treino. Ou seja, a rede tentará prever todo o conjunto de treino e, após isso, recalculará os pesos aplicados em toda a rede levando em consideração o erro cometido durante a iteração, com um método chamado *back propagation*.

As MLPs, portanto, não são capazes de armazenar informações de épocas anteriores, pois recalculam os pesos apenas com informações da iteração imediatamente anterior, o que faz com que para dados que possuam dependências de longo termo, é provável que o desempenho seja afetado por isso.

Então, para resolver esse problema, foram criadas redes neurais recorrentes (RNN), em que algumas MLPs são encadeadas, uma passando informações para a outra e os pesos são recalculados não só com base nos erros atuais, mas de acordo com estados anteriores da rede.

No entanto, podem ocorrer dois tipos de problemas nas RNNs: *Exploding Gradients* e *Vanishing Gradients*. Por conta deles, as redes LSTM [Hochreiter and Schmidhuber 1997] foram criadas, em 1997

Basicamente, esse tipo de rede neural funciona de modo parecido com as RNNs, mas possui um tratamento adicional para decidir quais informações são relevantes para o aprendizado e devem ser armazenadas e quais informações devem ser esquecidas. No modelo abaixo é possível visualizar a arquitetura teórica da rede LSTM. O chamado *forget gate* é onde ocorre a aplicação da função responsável por decidir esse tratamento das dependências de longo termo.

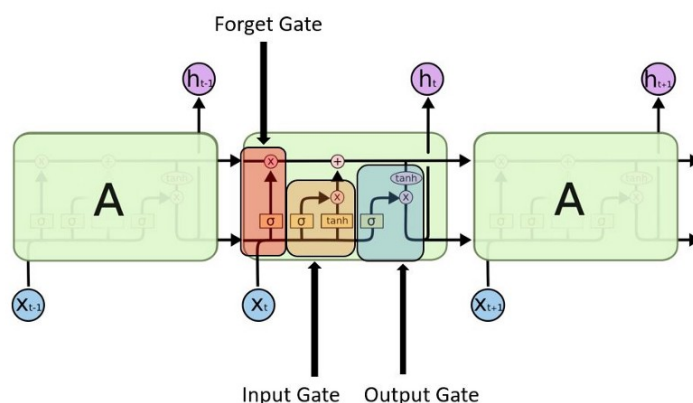


Figura 4. Long Short Term Memory (LSTM)

O problema da previsão de energia é um problema de série temporal, portanto a contextualização e o comportamento das variáveis ao longo é importante. Assim, a capacidade de selecionar e armazenar características relevantes relacionadas ao tempo, torna a LSTM uma ferramenta com potencial de fornecer um bom modelo preditivo para esse problema.

Sabendo do potencial supracitado e levando em que: "Redes Neurais Recorrentes são bastante afetadas por problemas como vanishing e exploding gradient, métodos de controle do gradiente e outros algoritmos de aprendizado devem ser investigados." [Heinen and Fernandes 2018]. A exploração dessa técnica foi alvo desse trabalho.

2.2.1. Medida de qualidade do modelo

A métrica utilizada para avaliar o modelo treinado foi a Raiz quadrada do Erro Quadrático Médio (em inglês, Root Mean Squared Error, ou simplesmente, RMSE), porque ela mede o quanto os valores obtidos estão errados em comparação aos esperados, por causa disso pode-se considerar o RMSE uma medida análoga ao desvio padrão [Zumel and Mount 2014]. Além deste ser a medida de erro usualmente usado para aferir qualidade de ajuste.

O RMSE é uma boa métrica a ser utilizada porque o modelo explorado no trabalho trata-se de uma técnica de previsão (regressão), então saber o quão certa está a previsão a partir desta medida é importante para poder avaliar os treinos e testes realizados. O RMSE é considerado uma métrica boa pela bibliografia porque ele explicitamente está usando o que a maioria dos mecanismos de avaliação tentam minimizar [Zumel and Mount 2014].

O RMSE é dado pela seguinte fórmula:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (d_i - f_i)^2}{n}}$$

onde n é o número de previsões, d_i é o valor previsto, f_i é o valor esperado.

2.2.2. LSTM aplicada ao conjunto de dados

O *dataset* escolhido pode ser considerado como uma série temporal porque são várias medições de uma hora sobre o estado do tempo, ordenadas de forma sequencial, portanto é importante que se mantenha essa ordem, porque ela é fundamental para entender o comportamento das variáveis e assim realizar a predição.

A LSTM é um modelo de aprendizado supervisionado [Hochreiter and Schmidhuber 1997], então precisa-se servi-la com dois conjuntos de dados para que ela possa realizar suas operações: o conjunto de entrada X e o de saída y . Considerando isso foi preciso realizar um tratamento no dados utilizados para deixar a execução em melhor performance de tempo, além de poder dividi-los nos conjuntos X e y requisitados. Este tratamento é chamado de janelamento.

O janelamento dos dados foi realizado considerando as variáveis contidas no recorte do *dataset*, a variável para ser prevista, o número de horas anteriores a serem observadas para fazer a previsão e quantas horas deseja-se prever. Com isso uma tabela com n linhas, em que n é o número de horas, e $(n_variaveis * h_anteriores) + h_previstas$ colunas é gerada. Seja t a hora atual, $t-1, t-2, \dots, t-m$ as horas anteriores e $t+1, t+2, \dots, t+o$ as horas futuras, cada linha da tabela possui o valor de todas as variáveis nas horas anteriores e o valores da variável a ser prevista nas horas adiante.

	RADIAÇÃO	TEMPERATURA	UMIDADE	RADIAÇÃO	TEMPERATURA	UMIDADE	RADIAÇÃO
	var1(t-2)	var2(t-2)	var3(t-2)	var1(t-1)	var2(t-1)	var3(t-1)	var1(t)
3	307.20	20.80	94.00	638.91	21.70	92.00	1151.18
4	638.91	21.70	92.00	1151.18	21.80	89.00	1033.75
5	1151.18	21.80	89.00	1033.75	20.80	95.00	903.77
6	1033.75	20.80	95.00	903.77	21.70	99.00	1922.81
7	903.77	21.70	99.00	1922.81	22.80	88.00	469.23
8	1922.81	22.80	88.00	469.23	20.70	98.00	157.89
9	469.23	20.70	98.00	157.89	20.40	97.00	171.18

Figura 5. Exemplo de janelamento

A figura 5 é um exemplo de janelamento utilizando uma parte simplificada do conjunto de dados trabalhado em que tem apenas três variáveis, com base nelas e nas duas horas anteriores deseja-se prever a radiação solar na hora atual.

Posteriormente, é feita a separação desta tabela em conjuntos de entrada X e saída y , neste caso, y é composto por todos os valores da hora que deseja-se realizar a previsão

e X possui todos os dados de todas as variáveis nos tempos $t - m$ até $t - 1$. Então os conjuntos X e y são divididos em conjuntos de treino e teste baseado na proporção que será explicada na próxima seção, resultando, assim, em quatro conjuntos: $train_X$, $train_y$, $test_X$ e $test_y$.

```
# design network
model = Sequential()
model.add(LSTM(n_neurons, input_shape=(
    train_X.shape[1], train_X.shape[2]), activation='tanh', recurrent_activation='hard_sigmoid', use_bias=u_bias))
model.add(Dense(1))
# fit network
history = model.fit(train_X, train_y, epochs=100, batch_size=75, validation_data=(
    test_X, test_y), verbose=2, shuffle=False)
```

Figura 6. Código em python que mostra os parâmetros que as funções "LSTM" e *fit* do keras recebem

A figura 6 mostra quais parâmetros são usados para configurar o modelo e quais são os que são usados para realizar o treino do modelo. Para os parâmetros do modelo tem-se o número de neurônios, o formato da entrada, as funções de ativação, o uso do bias e quantas camadas escondidas. Para os parâmetros do treino são passados os conjuntos de treino, o número de épocas, o tamanho do lote de dados que serão treinados em uma iteração, os dados de validação(conjunto de testes), como deve ser feita a demonstração do progresso e se deve treinar de forma sequencial ou aleatória, isto é, usando horas separadas umas das outras.

2.2.3. Escolha dos parâmetros

A escolha dos parâmetros fixos e variáveis a seguir, foi feita com base nos resultados da execução da LSTM para a estação de São Paulo. Além dos parâmetros padrão da LSTM (camadas escondidas, funções de ativação e tamanho do *batch*), àqueles que apresentaram grande variação do RMSE foram fixados (e.g. número de neurônios), sendo selecionado os que possuíam melhor valor (menor erro). Outro parâmetro fixado foi o *shuffle* porque, acredita-se que não faria sentido executar um treinamento em dados do conjunto de forma aleatória no contexto de previsão de séries temporais, porque a ordem dos acontecimentos é extremamente relevante para o problema.

Quanto às variáveis, foram elegidas quantas horas precedentes seriam utilizadas para previsão e quantas horas posteriores seriam previstas. Os valores 14 e 98 representam uma semana e um mês, respectivamente.

1. Parâmetros variáveis

- Atributos A, B e C definidos na seção 2.1.3
- Divisão do *dataset* nas proporções de treino/teste: (75%, 25%), (67%, 33%), (50%, 50%), (33%, 67%) e (25%, 75%);
- Horas passadas utilizadas para previsão : 3, 14 e 98
- Horas previstas 1, 3, 7 e 14

2. Parâmetros fixos :

- Camadas escondidas: 1;
- Neurônios: 50;
- Épocas: 100;

- Funções de ativação: Tahn (tangente hiperbólica) e *Hard sigmoid*;
- Tamanho do batch: 75;
- *shuffle*: *false*;

3. ANÁLISE DOS RESULTADOS

Para a confecção dos testes, cada um dos atributos foi permutado com os outros três parâmetros variáveis e o RMSE foi armazenado em uma tabela. Cada teste foi nomeado da forma : Atributo_HorasPassadas_HorasPrevistas. A figura 7 representa as melhores (erro mínimo) e piores (erro máximo) variações de cada estação, por atributo.

Resultados											
MELHORES						PIORES					
ALFREDO						ALFREDO					
A		B		C		A		B		C	
MIN	0.67	MIN	0.67	MIN	0.67	MAX	0.25	MAX	0.33	MAX	0.33
	359.66		349.91		342.62		1033.60		993.19		991.61
	A 14 1		B 98 1		C 98 1		A 3 7		B 3 7		C 3 7
IBITINGA						IBITINGA					
A		B		C		A		B		C	
MIN	0.67	MIN	0.67	MIN	0.67	MAX	0.25	MAX	0.25	MAX	0.25
	375.81		366.17		349.64		1168.04		1051.86		1004.06
	A 98 1		B 98 1		C 14 1		A 3 7		B 3 7		C 3 7
SÃO GONÇALO						SÃO GONÇALO					
A		B		C		A		B		C	
MIN	0.33	MIN	0.33	MIN	0.33	MAX	0.33	MAX	0.50	MAX	0.50
	252.26		249.32		235.90		696.19		626.13		568.29
	A 98 1		B 98 1		C 98 1		A 3 3		B 3 3		C 3 7
SÃO PAULO						SÃO PAULO					
A		B		C		A		B		C	
MIN	0.75	MIN	0.75	MIN	0.75	MAX	0.25	MAX	0.25	MAX	0.25
	372.23		367.61		375.36		1092.00		1061.04		1042.66
	A 98 1		B 98 1		C 14 1		A 3 7		B 3 7		B 3 7

Figura 7. Melhores e piores resultados para cada estação

Observando os resultados, percebe-se que a maioria dos melhores desempenhos foram obtidos em conjuntos com um volume maior de dados para teste do que treino, sendo 67% e 75% utilizados para treino. No entanto, a estação de São Gonçalo, que teve o menor dos erros, teve seus melhores resultados concentrados em 33% de teste.

Sabendo que o recorte é sequencial e não-aleatório, pois as relações de tempo são importantes, supõe-se que esse fenômeno ocorra, provavelmente, pelo período em que o modelo está sendo treinado (final do ano) e as particularidades de cada estação.

Além disso, é notável que todos os melhores resultados foram aqueles em que se buscava prever apenas a próxima hora. Isso indica que quanto menor o número de horas a serem previstas, maior a facilidade que o modelo tem para acertar.

Outro ponto a ser considerado é o de que as horas utilizadas para previsão foram um fator relevante, pois de maneira geral, quanto maior essa quantidade, mais baixos foram os erros, apesar de que em alguns casos, utilizando apenas um dia (14 horas) anterior a hora prevista o erro foi menor que o mesmo teste realizado com uma semana (98 horas). Mas a tendência observada é reforçada pelo fato de que todos os piores resultados se concentram nos testes com menor número de horas utilizadas para previsão.

Portanto, para verificar a hipótese de que quanto maior o número de horas utilizadas na previsão, menor o erro, seria necessário estender a quantidade de horas para um

número maior, um ou mais meses, testes que exigiriam maior poder computacional e/ou tempo.

Olhando para os piores resultados, foi possível perceber outras tendências. Como os melhores resultados concentravam-se nos teste de previsão da próxima hora, esperava-se que os piores resultados estivessem concentrados no número máximo de horas a serem previstas, 14 horas. No entanto, a maioria dos resultados se concentrou em previsões das 7 horas futuras.

Esse comportamento contraintuitivo está exemplificado na figura abaixo, mas fora observado em todas as estações. Uma possível explicação para isso é a de que o padrão comportamental da incidência de energia solar ao longo de um dia é mais atenuado e previsível do que meio dia.

ALFREDO																	
A						B						C					
	0.25	0.33	0.50	0.67	0.75		0.25	0.33	0.50	0.67	0.75		0.25	0.33	0.50	0.67	0.75
A_3_7	1168.04	1067.82	889.32	841.27	1106.95	B_3_7	1051.86	985.01	883.24	822.86	987.56	C_3_7	1004.06	963.09	853.26	809.23	940.85
A_3_14	764.87	729.36	651.93	609.96	602.18	B_3_14	770.17	727.79	656.34	611.14	604.21	C_3_14	788.23	736.92	679.33	612.05	607.20
A_14_7	759.09	765.48	639.51	611.87	805.67	B_14_7	763.35	724.54	643.21	650.22	646.08	C_14_7	753.90	749.12	656.77	645.00	630.47
A_14_14	736.59	750.36	651.05	600.16	649.59	B_14_14	733.42	738.04	661.22	606.22	646.81	C_14_14	734.01	718.99	615.00	596.92	607.24
A_98_7	788.90	694.28	660.32	588.86	692.31	B_98_7	775.04	719.99	624.96	611.85	686.30	C_98_7	737.59	723.85	637.07	621.43	663.17
A_98_14	774.18	745.80	624.05	606.10	648.27	B_98_14	740.01	707.60	620.08	590.35	679.99	C_98_14	755.12	703.74	615.71	577.91	662.94

Figura 8. Erros 7 e 14 horas para a estação de Alfredo

Além da análise dos erros finais, foi necessário observar o comportamento do erro ao longo do treinamento do modelo a fim de avaliar se o número de épocas escolhido foi suficiente para a convergência e identificar possíveis indícios de *overfitting* ou *underfitting*.

Observando o exemplo de *overfitting*, percebe-se que o erro no teste diminui até certo ponto e começa a elevar-se no final, enquanto que o comportamento do erro durante o treino é estabilizou depois de um determinado número de épocas. Isso ocorre devido ao fato do modelo ter sido bem ajustado aos dados do treino, mas esse modelo não conseguir generalizar para os dados do teste.

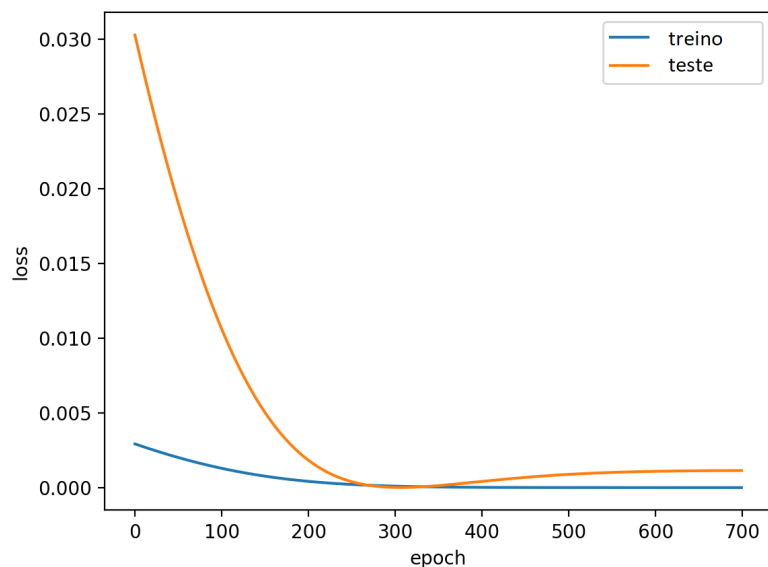


Figura 9. Gráfico de diagnóstico com *overfitting*

Já no caso de um *underfitting*, haveria um distanciamento dos erros nos testes em relação aos do treino, sendo que os erros nos testes seriam consideravelmente maiores que os erros no conjunto de treino. A diferença gráfica entre os dois problemas é de que os erros no testes estariam mais distantes no *underfitting* e não haveria degradação como há no *overfitting*.

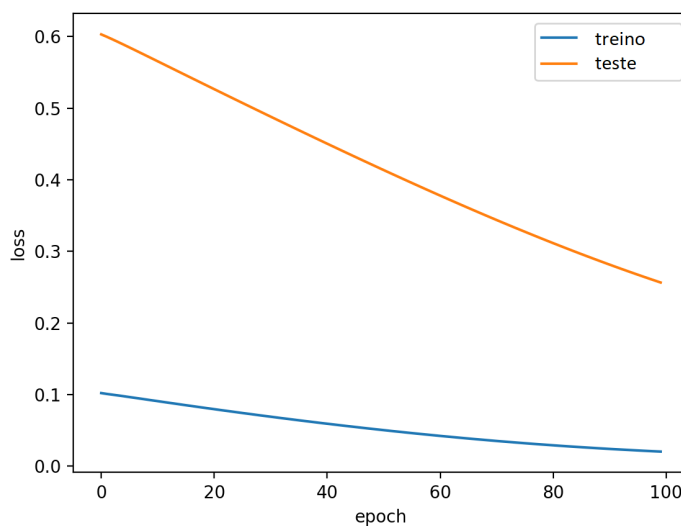


Figura 10. Gráfico de diagnóstico com *underfitting*

Selecionando alguns dos melhores e piores resultados para analisar o erro, observou-se que o comportamento nos melhores não indicava nenhum dos dois problemas, mas nos piores resultados havia indícios de *overfitting*. Portanto, a etapa final de testes consistiu em combinar os modelos de estações.

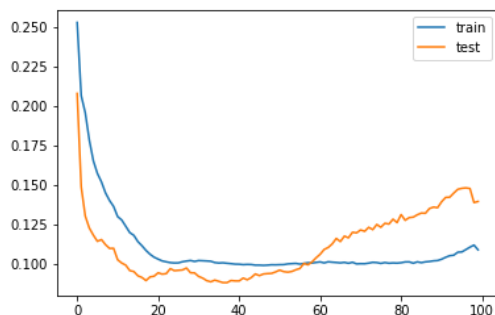


Figura 11. Gráfico de diagnóstico dos erros no pior caso: conjunto de parâmetros A, 3 horas anteriores, 3 horas previstas e 33% teste

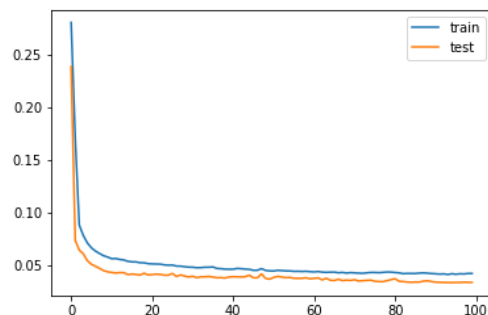


Figura 12. Gráfico de diagnóstico dos erros no melhor caso: conjunto de parâmetros C, 98 horas anteriores, 1 hora prevista e 33% teste

A combinação se deu da seguinte maneira: foram selecionados os melhores parâmetros para cada estação de acordo com os menores erros; foram repetidos os treinos com esses parâmetros e, além de realizar testes com a estação treinada, utilizou-se modelos de outras estações para testar. O resultado desses testes pode ser observado na tabela abaixo.

Alfredo - 98 - 1 - 0.67 - C		Ibitinga - 14 - 1 - 0.67 - C		São Gonçalo - 98 - 1 - 0.33 - C		São Paulo - 98 - 1 - 0.75 - C	
Alfredo	339.60	Alfredo	353.00	Alfredo	409.37	Alfredo	358.86
Ibitinga	364.93	Ibitinga	332.48	Ibitinga	417.39	Ibitinga	340.09
SaoGoncalo	269.70	SaoGoncalo	274.65	SaoGoncalo	243.49	SaoGoncalo	289.00
SaoPaulo	387.81	SaoPaulo	389.65	SaoPaulo	485.29	SaoPaulo	369.41

Figura 13. Tabela dos erros de combinações de estações

Mais uma vez, a estação de São Gonçalo foi notável. Os dados dessa estação quando previstos por outros modelos tiveram melhores desempenhos que todas as outras estações. Isso significa que os dados obtidos na estação de São Gonçalo para o ano testado são mais previsíveis que para outras estações. Outro indício disso se dá pelo fato de que o modelo com melhores previsões não é o de São Gonçalo. É como se o treino em São Gonçalo fosse "fácil" e não permitisse acertar bem o comportamento do clima em outras estações, mas a previsibilidade dos dados dela permite que sejam mais facilmente previstos.

Olhando para o desempenho do algoritmo nas primeiras 14 horas (um dia) do treino realizado na estação de Ibitinga – modelo cuja somatória dos erros foi a menor – e teste realizado com São Gonçalo, percebe-se que o ajuste das previsões do modelo ao comportamento dos dados fez com que essas fossem bem próximas das medidas reais, pois os picos e vales coincidem sempre e as diferenças, apesar de perceptíveis, são relativamente baixas. O RMSE não foi tão baixo quanto o melhor resultado nos testes anteriores, mas considerando que são estações diferentes, é uma diferença aceitável.

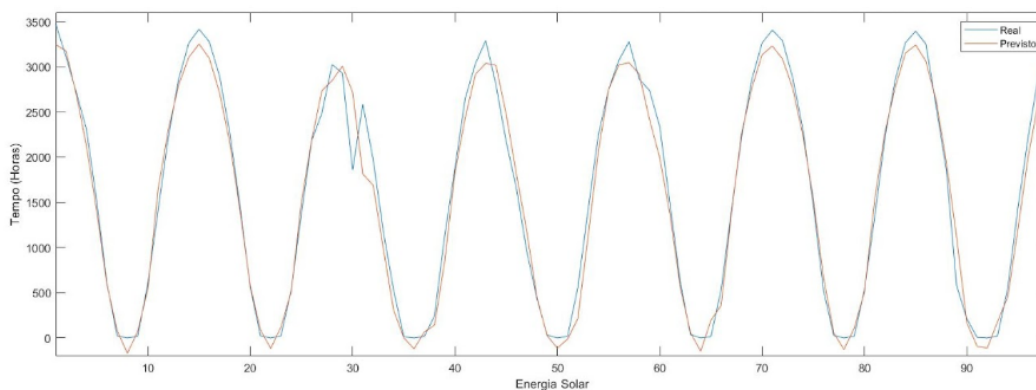


Figura 14. Previsto x Real: Modelo Ibitinga com a validação em São Gonçalo

4. CONCLUSÕES

A combinação de parâmetros e análise dos erros permitiu a identificação de algumas tendências relevantes. Quanto maior o número de horas utilizadas para realizar a previsão, melhor a capacidade do modelo de se ajustar, no entanto, seriam necessários testes com um número ainda maior de horas, pois os melhores resultados se concentraram em 98 e 14 horas e todos os piores resultados eram os que utilizavam o mínimo de horas (3) para realizar previsões.

Esperava-se que o pior desempenho ocorresse na previsão de 14 horas, pois era o maior número que buscava-se prever. No entanto, observou-se um comportamento inesperado, em que os erros de previsões para 7 horas eram maiores, em quase todos os testes. Esse comportamento poderia ser explicado pelo fato de que a energia solar flui ao longo do dia em um comportamento específico, mais previsível do que em intervalos de meio dia. Para validar essa hipótese seria necessário um conjunto de dados com medidas de 24 horas completas.

A suspeita de *overfitting* por conta da comparação entre os erros de treino e teste ao longo da construção do modelo nos piores casos, foi verificada por meio de uma combinação de estações, em que os melhores modelos de cada estação eram utilizados para fazer previsões com dados de outras estações. Analisando as previsões desses testes, por meio da comparação entre valor previsto e valor real, considerou-se que o desempenho do algoritmo na previsão foi satisfatório e o modelo treinado em uma estação poderia ser reutilizado em outras estações.

Assim, a rede LSTM foi considerada uma técnica viável para a tarefa de previsão de energia solar, mas com algumas ressalvas de que seria necessário confirmar as tendências observadas para refinar os parâmetros do modelo e adquirir um desempenho mais satisfatório.

5. TRABALHOS FUTUROS

A partir dos resultados obtidos nesse trabalho, é possível elaborar uma estratégia de testes que possam refinar os resultados dos algoritmos. Uma linha de pesquisa seria verificar as tendências observadas e trazer modelos melhores para os dados aqui utilizados.

Outra possibilidade seria a de aumentar o escopo dos testes para outros anos e/ou estações e trazer novas ideias de parametrização do algoritmo.

Sabendo que o modelo pode ter um bom desempenho quando treinado em uma estação e testado em outras, provavelmente o principal foco de um próximo estudo seria observar em maior profundidade como os resultados se comportariam com essa combinação de estações, pois melhorar o desempenho nessa tarefa seria o que melhor viabilizaria uma aplicação dessa técnica.

6. REFERÊNCIAS BIBLIOGRÁFICAS

Referências

Araújo, A., Castro, U. R. M. d., Silva, L. H. d. C., Lima, T., Assunção, F., and LNP, C. (2018). Hourly weather surface - brazil (southeast region).

Google (2015). Project sunroof. <https://www.google.com/get/sunroof#p=0/>.

Heinen, E. D. and Fernandes, R. A. S. (2018). Redes neurais recorrentes aplicadas à previsão de radiação solar no horizonte de curto prazo. *Computer on the Beach*, 9:841–849.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Lipton, Z. C., Elkan, C., Kale, D. C., and Wetzel, R. (2016a). Clearing to diagnose with lstm recurrent neural networks. *International Conference on Learning Representations (ICLR)*, 56.

Lipton, Z. C., Kale, D. C., and Wetzel, R. (2016b). Modeling missing data in clinical time series with rnn. *Proceedings of Machine Learning for Healthcare*, 56.

Program, U. . N. D. and Council, W. . E. (2000). *The T_EX Book*. Addison-Wesley, 15th edition.

Zumel, N. and Mount, J. (2014). *Practical Data Science with R*. Manning, 1st edition.