Deep Learning sobre dados meteorológicos tabulares: Relatório Final

Raysa Masson Benatti
176483
raysa.benatti@gmail.com

Mauricio de Sousa Araujo
184477
m184477@dac.unicamp.br

Matheus de Souza Ataide
147375
email address

Ao longo da última década, modelos de deep learning têm sido cada vez mais usados e estudados para resolver problemas em diferentes domínios. Em geral, tais modelos têm boa performance sobre dados não estruturados, como imagens, áudio e texto. Embora menos explorado, seu uso sobre dados tabulares também é promissor. Neste trabalho, usamos técnicas de deep learning para abordar o problema de previsões em séries temporais de dados meteorológicos — domínio cuja demanda por estudos tem crescido em razão da emergência climática. Mostramos como diferentes arquiteturas de redes recorrentes — em particular, redes LSTM e redes baseadas em Transformer — podem ser adequadas para efetuar tal tarefa, além de refletir sobre algumas de suas limitações.

I. Introdução

Técnicas de deep learning têm sido crescentemente exploradas na última década em razão do aumento de poder de processamento computacional e disponibilidade de dados, o que permitiu o uso e desenvolvimento de diferentes arquiteturas de redes neurais profundas. Um dos maiores trunfos dessas ferramentas é a possibilidade de reconhecer padrões em dados não estruturados, como imagens, sinais elétricos e texto. Em razão disso, seu uso tem sido concentrado em áreas como visão computacional e processamento de linguagem natural.

Menos explorado tem sido o uso de tais técnicas sobre dados tabulares ou estruturados. Em geral, métodos tradicionais de Machine Learning performam melhor nesse universo, sendo, portanto, mais utilizados. No entanto, o interesse em experimentar ferramentas de aprendizado de máquina profundo sobre esse tipo de dado vem aumentando — afinal, parte considerável dos dados disponíveis em diversos domínios está em formato tabular.

Embora, em geral, não exista estrutura hierárquica a ser explorada em dados estruturados, há indícios de que técnicas de deep learning podem agregar vantagens ao processo — especialmente para datasets maiores —, como facilitar eventuais análises multimodais e análise de dados online, e diminuir a necessidade por *feature engineering* [1, 2].

Nesse cenário, diversas arquiteturas têm sido exploradas, sobre muitas aplicações [3, 4, 5, 6, 7, 8]. Em particular, estamos interessados em técnicas de previsão em séries temporais: dados tabulares históricos são fartamente disponíveis e relativamente fáceis de obter, em diversos domínios. O domínio no qual focamos, abarcado pela motivação de *Artificial Intelligence for Social Good* [9] e cuja importância tem crescido notavelmente, é o da exploração de dados climáticos.

O clima tem impactos significativos sobre diversas atividades humanas e setores econômicos. Além disso, mudanças climáticas são uma realidade cada vez mais preocupante, diante de sua ameaça à vida humana e à biodiversidade. Tradicionalmente, o clima tem sido objeto de estudo de ciências como a física e a geografia; recentemente, técnicas de aprendizado de máquina têm se somado às ferramentas usadas para estudá-lo.

Nesse contexto, é importante entender como diferentes modelos preditivos se comportam sobre dados desse domínio e quais parâmetros são importantes para sua implementação. Aliando essa motivação aos objetivos da disciplina, buscamos, nesse trabalho, explorar a aplicação de modelos de deep learning na área.

O presente relatório se organiza da seguinte forma: a subseção I.A descreve os objetivos do trabalho. A Seção II detalha a metodologia adotada: dados, arquiteturas de rede e detalhes relevantes de implementação. A Seção III exibe os resultados e discussões pertinentes. Por fim, a Seção IV traz considerações finais e sugestões para trabalhos futuros.

O código cuja implementação este relatório descreve pode ser acessado aqui.

A. Objetivos

O objetivo do trabalho é verificar a adequação de modelos de deep learning para efetuar previsões em séries temporais de dados climáticos tabulares reais. A partir dessa meta, desenham-se os seguintes objetivos específicos:

- Definição e coleta de um conjunto de dados climáticos tabulares;
- Definição e implementação de modelos de deep learning adequados para previsão em séries temporais;
- Definição e cálculo de métricas e parâmetros de avaliação;
- Comparação entre resultados e sua sistematização.

II. Metodologia

A metodologia do trabalho baseou-se em três grandes etapas: (a) coleta, exploração e pré-processamento de dados; (b) aplicação de modelos de redes neurais recorrentes LSTM

(*long short-term memory*); (c) aplicação de um modelo de rede baseado em Transformer.

A. Dados

O objetivo original do grupo era trabalhar com o conjunto de dados climáticos disponibilizado pelo INMET (Instituto Nacional de Meteorologia). Esse objetivo foi abandonado; conforme descrito no *baseline*, a exploração inicial revelou a impossibilidade de aplicar os modelos sobre esse conjunto, preenchido com muitos valores NaN *(not a number)*. Optou-se, então, por usar o arquivo de valores absolutos de métricas climáticas mensais da estação convencional do Posto Meteorológico de Piracicaba, SP [10].

Esse conjunto traz, para todos os meses desde 1917, os valores de onze *features* medidas na estação, dentre as quais selecionamos **temperaturas máximas** e **temperaturas mínimas**. O arquivo tem 1235 instâncias: uma para cada um dos doze meses, durante 103 anos, exceto para 2020 (que tem registros até novembro).

Os dados passaram por pré-processamento para organizá-los cronologicamente, de modo a exibir registros antigos sempre antes de registros novos. O trabalho incluiu, ainda, renomear colunas, converter tipos e deletar instâncias com valores NaN, para viabilizar as operações.

Por fim, os dados foram separados em conjuntos de treino, validação e teste, na proporção 70:20:10, levando-se em conta a dependência temporal.

B. LSTM

LSTM, ou *long short-term memory*, são redes neurais recorrentes usadas para processar dados sequenciais. A implementação das redes LSTM adotadas neste trabalho seguiu a sugestão elaborada por Jason Brownlee [11]. O autor descreve sete modelos LSTM para previsão em séries temporais, dentre os quais selecionamos dois: *vanilla*, ou básico, e bidirecional.

No *baseline*, exploramos, somente para uma variável (temperatura máxima), duas configurações da rede básica: com 15 e 50 unidades LSTM. Os demais parâmetros foram mantidos: ReLU como função de ativação, Adam como otimizador, erro quadrático médio (MSE) como métrica de erro e 200 épocas. O modelo inclui, ainda, definir janelas deslizantes de instâncias, com tamanho customizável, representado pela variável `n_steps` (no nosso experimento inicial, usamos janelas de 10 valores de entrada para cada saída).

A Figura 1 ilustra o conceito de janelas deslizantes para uma janela de tamanho 5. A série do exemplo tem 11 valores. Cada entrada de 5 valores (tamanho da janela) tem uma saída (previsão) correspondente, que é o valor imediatamente a seguir; a cada iteração, a janela desliza, passando a iniciar um valor acima. O deslocamento da janela se repete até que não haja mais saídas possíveis.

A arquitetura da rede LSTM básica consiste em duas camadas, sendo a primeira conectando as entradas $x_1, ..., x_n$ com as unidades LSTM, e a segunda conectando estas unidades com um neurônio de saída Σ, ativado por uma função φ

Figure 1. Representação do conceito de janelas deslizantes. Em vermelho, valores da janela; em azul, valor de saída da janela correspondente.

(ReLU, no nosso caso). As duas versões — com 15 e 50 unidades — foram treinadas sobre os dados de treino e avaliadas graficamente, com plotagem da evolução de *loss* (MSE) e de previsões sobre o conjunto de validação. Os resultados dos experimentos iniciais foram apresentados no *baseline*. A Figura 2 ilustra a arquitetura da rede LSTM *vanilla*, em que Y representa a saída.

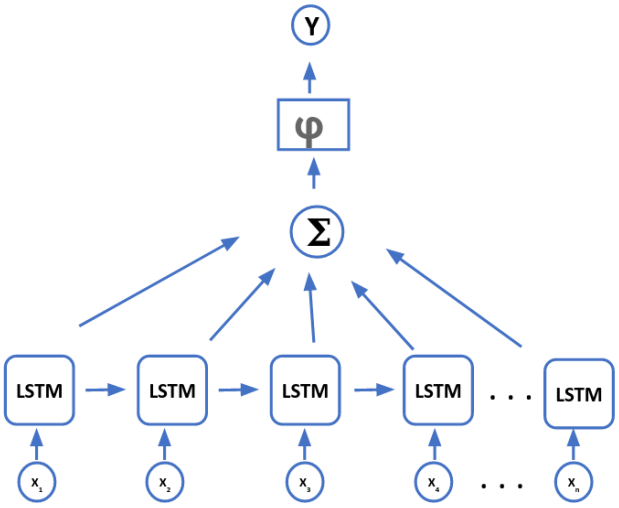

Figure 2. Representação da rede LSTM básica.

A partir da exploração inicial, as seguintes modificações foram estabelecidas para a versão final do projeto:

- Opção pelo uso de redes de 15 unidades (que performaram melhor que com 50 unidades);
- Cálculo de erro médio absoluto (MAE), além do MSE;
- Cálculo do tempo de execução;
- Realização de testes com três tamanhos de janela: 3, 10 e 20;
- Redução de 200 para 100 épocas;
- Realização de experimentos para as duas variáveis de interesse;

- Implementação de uma rede LSTM bidirecional;
- Implementação de um modelo baseado em Transformer (detalhado na Seção II.C).

A rede LSTM bidirecional permite que o modelo aprenda a sequência usada para treino nos dois sentidos (”de frente pra trás” e ”de trás pra frente”), conjugando ambas as interpretações [11]. O objetivo é, então, verificar se isso se traduz em melhor previsão no nosso cenário.

C. *Transformer*

O modelo Transformer foi apresentado no artigo *Attention is All You Need* [12], que impulsionou um progresso recente significativo na área de Processamento de Linguagem Natural (PLN). Tal modelo trouxe consigo um mecanismo conhecido como mecanismo de atenção. A partir do sucesso dessas redes em PLN, adaptações passaram a ser sugeridas e exploradas em problemas como previsão em séries temporais [13]. Afinal, ambos envolvem o processamento de dados sequenciais. A implementação das redes Transformer adotadas nesse trabalho seguiu o código elaborado por Husein Zolkepli [14].

A Figura 3 mostra a estrutura da rede utilizada nesse trabalho. As seguintes configurações foram estabelecidas para a versão final do projeto:

- 8 *heads* de *attention*;
- Camadas *Dense* com 512 neurônios e ReLU como função de ativação;
- Camadas lineares com 128 neurônios;
- Utilização de janelas deslizantes de tamanhos 3, 10 e 20;
- 100 épocas;
- Utilização de *Posicional Encoding*, como sugerido no artigo original [12].

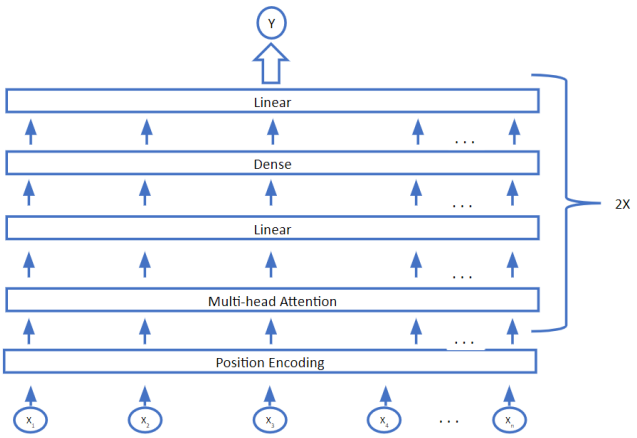

Figure 3. Representação da rede com mecanismo de atenção.

III. Resultados e Discussão

A. *LSTM*

A Tabela I resume os resultados observados para a LSTM *vanilla*. As células azuis representam os experimentos para temperaturas mínimas, e as células vermelhas representam os experimentos para temperaturas máximas. As células cinza mostram valores de média aritmética (M) e desvio-padrão (DP) das variáveis de cada coluna. O mesmo se aplica à Tabela II, que resume os resultados observados para a LSTM bidirecional. Os valores de erro referem-se, sempre, ao conjunto de validação.

Janela	MAE	MSE	Tempo de execução (s)
3	2.09	6.59	10.39
10	2.04	6.28	16.90
20	2.09	6.65	26.51
	M=2.07	M=6.51	M=17.93
	DP=0.02	DP=0.16	DP=6.62
3	1.83	5.70	10.11
10	1.43	4.49	17.17
20	1.25	2.84	27.32
	M=1.50	M=4.34	M=18.20
	DP=0.24	DP=1.17	DP=7.06

Table I
RESULTADOS PARA LSTM BÁSICA

Janela	MAE	MSE	Tempo de execução (s)
3	2.08	6.57	14.27
10	2.12	6.67	23.73
20	1.66	4.43	37.14
	M=1.95	M=5.89	M=25.05
	DP=0.21	DP=1.03	DP=9.38
3	1.31	3.13	13.71
10	1.24	2.76	24.98
20	1.19	2.56	37.30
	M=1.25	M=2.82	M=25.33
	DP=0.05	DP=0.24	DP=9.63

Table II
RESULTADOS PARA LSTM BIDIRECIONAL

Em todos os casos, aumentar o tamanho das janelas aumentou o tempo de execução, o que é esperado. O tempo de execução da LSTM bidirecional foi sempre maior que o da LSTM básica, e não diferiu significativamente entre prever temperaturas máximas e temperaturas mínimas.

A LSTM bidirecional se mostrou mais adequada para realizar as previsões, com valores de erro menores. É interessante notar que, para temperaturas mínimas, o desvio do erro (para janelas de diferentes tamanhos) é maior na LSTM bidirecional; para temperaturas máximas, o desvio do erro é maior na LSTM básica. Ou seja: para prever temperaturas mínimas, mudar o tamanho das janelas parece fazer menos diferença para a LSTM básica que para a bidirecional — no caso das temperaturas máximas, ocorre o contrário.

Além disso, os erros são menores para previsão de temperaturas máximas que de temperaturas mínimas; a discrepância é reforçada pelo fato de que as temperaturas máximas são, naturalmente, maiores. Detectamos, ainda, uma tendência de *overfitting* dos modelos para previsão de temperaturas mínimas, o que não ocorreu com temperaturas máximas. A Figura 4 ilustra isso: na convergência, a *loss* sobre o conjunto de validação (azul) é maior e oscila mais que a *loss* sobre o conjunto de treino (laranja).

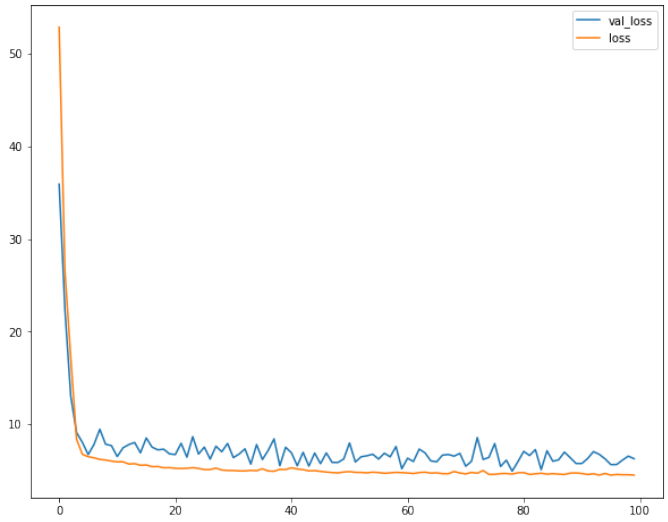

Figure 4. Evolução da *loss* da LSTM básica com janelas de 10 valores para previsão de temperaturas mínimas.

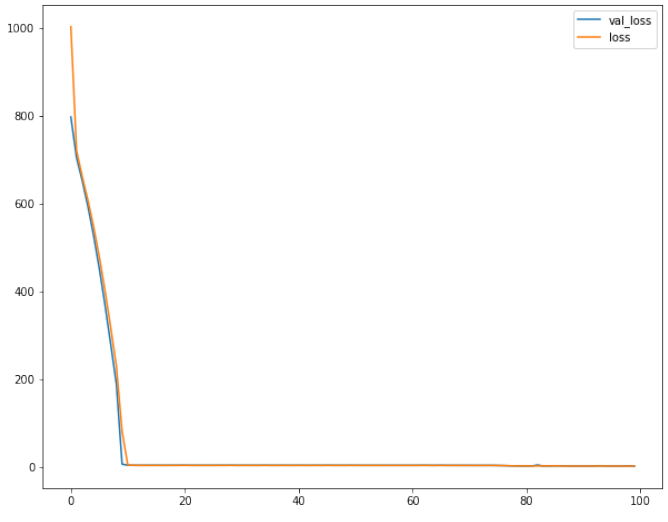

Figure 5. Evolução da *loss* da LSTM básica com janelas de 20 valores para previsão de temperaturas máximas.

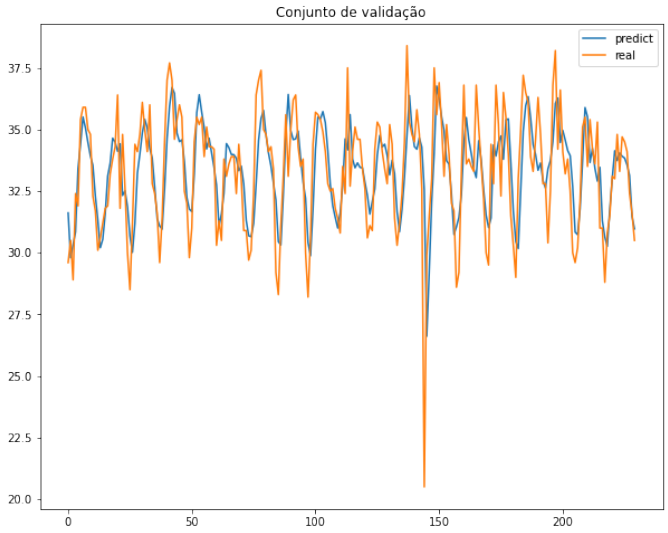

Figure 6. Dados reais e previsões de temperaturas máximas da LSTM básica com janelas de 20 valores.

Notamos, também, que aumentar a quantidade de janelas, em geral, melhora a qualidade da previsão — exceto para a previsão de temperaturas mínimas com a LSTM básica, em que o modelo com janelas de tamanho 10 foi o que melhor performou. Uma hipótese explicativa é a de que, nesse caso, a janela maior introduziu ruídos na análise, que não apareceram — ou foram eliminados — na LSTM bidirecional.

Selecionamos, a seguir, quatro gráficos que ilustram o desempenho das redes. As Figuras 5 e 6 exibem, respectivamente, a evolução da *loss* e os valores — reais e previstos — para a melhor rede LSTM básica: a de janelas de 20 valores para previsão de temperaturas máximas. As Figuras 7 e 8, por sua vez, exibem o mesmo para a melhor rede LSTM bidirecional. O conjunto completo de gráficos produzidos nos experimentos pode ser acessado aqui.

Em todos os gráficos que mostram séries, a linha azul representa os valores previstos pelo modelo e a linha laranja representa os valores reais — sempre em relação ao conjunto de validação. Em todos os gráficos que mostram evolução da *loss*, a linha azul representa o resultado do conjunto de validação, e a linha laranja, o resultado do conjunto de treino.

Os gráficos exibem padrões semelhantes em todos os experimentos e mostram que as redes convergem rápido, em menos de 20 épocas. A LSTM bidirecional aprende mais rápido que a LSTM básica. Além disso, apesar dos resultados satisfatórios, que comprovam a adequação das redes escolhidas para realizar previsões em séries temporais, quase todos os modelos falharam em prever valores extremos. Isso é discutido em detalhes na Seção III.D.

B. *Transformer*

A Tabela III resume os resultados observados para a Transformer. As mesmas observações feitas para as Tabelas I e II são válidas aqui. O tempo de execução se repete para a previsão das duas variáveis pois, nesse caso, calculamos o tempo total de treinamento de cada modelo, para ambas — o resultado da tabela é a divisão desse valor por dois.

O tempo de treinamento é consideravelmente maior para a Transformer que para as LSTM e, ao contrário do que ocorre com as LSTM, aqui esse tempo **diminui** conforme aumenta o tamanho das janelas. Em todos os casos da nova rede, aumentar o tamanho das janelas também aumentou a qualidade da previsão.

Aqui, também, a previsão de temperaturas máximas foi melhor que a previsão de temperaturas mínimas; a última ainda se revelou mais sensível à mudança de tamanho das janelas. Todos os valores de erro da Transformer foram maiores que os das redes LSTM, exceto um: o MAE para janelas de tamanho 20 em previsão de temperaturas mínimas. Essa diferença, no entanto, não é significativa; de modo geral, a nova rede não

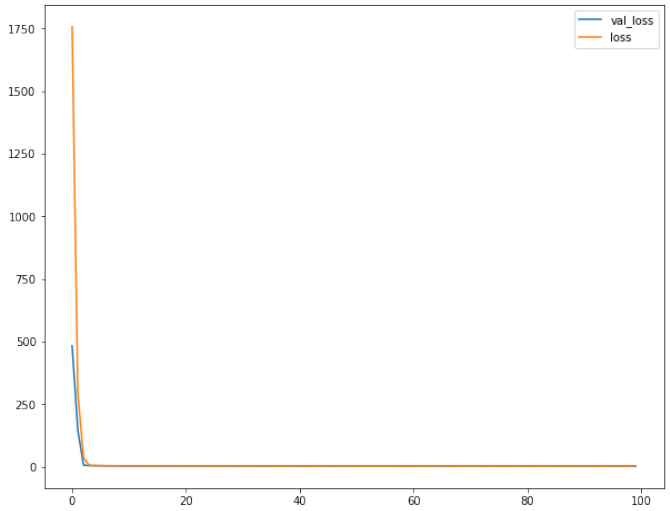

Figure 7. Evolução da *loss* da LSTM bidirecional com janelas de tamanho 20 para previsão de temperaturas máximas.

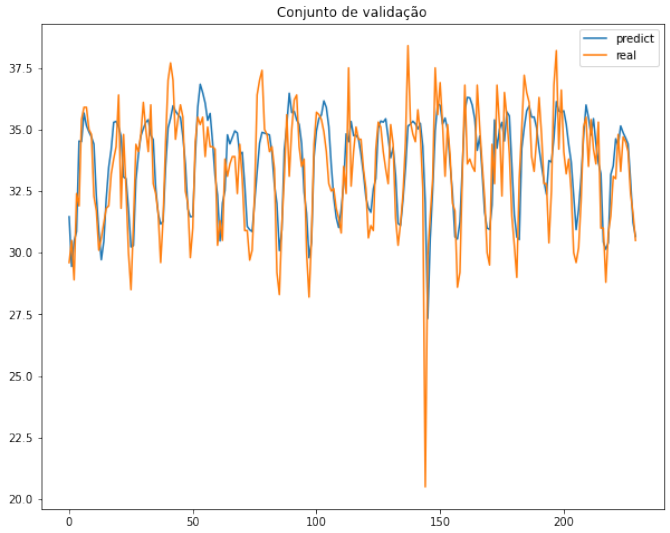

Figure 8. Dados reais e previsões de temperaturas máximas da LSTM bidirecional com janelas de tamanho 20.

performou melhor que as LSTM, mesmo apresentando um resultado coerente.

As Figuras 9, 10 e 11 ilustram os resultados da Transformer para janelas de tamanho 3, 10 e 20, respectivamente. Em todas elas, a linha preta representa os dados reais de temperaturas máximas, e a linha azul, a previsão do modelo — sempre em relação ao conjunto de validação. Pelos gráficos, é ainda mais visível a evolução da performance com o aumento do tamanho das janelas: embora, em todos os casos, a rede consiga captar a natureza periódica dos dados, a previsão tem muito mais qualidade quando se usam janelas maiores. O mesmo ocorre com a previsão de temperaturas mínimas. Notamos que aqui, também, o problema acerca da previsão de valores extremos permanece. O conjunto completo de gráficos produzidos nos experimentos pode ser acessado aqui.

Janela	MAE	MSE	Tempo de execução (s)
3	6.60	55.99	171.38
10	4.12	23.20	65.09
20	2.01	6.18	38.28
	M=4.24	M=28.46	M=91.58
	DP=1.88	DP=20.67	DP=57.48
3	2.22	7.99	171.38
10	2.50	8.57	65.09
20	1.33	3.08	38.28
	M=2.02	M=6.55	M=91.58
	DP=0.50	DP=2.46	DP=57.48

Table III
RESULTADOS PARA TRANSFORMER

Figure 9. Dados reais e previsões de temperaturas máximas da Transformer com janelas de tamanho 3.

C. *Teste final*

O teste final consistiu em aplicar o melhor de todos os modelos, de acordo com os experimentos acima, sobre o conjunto de teste. No nosso caso, esse modelo foi a rede LSTM bidirecional com janelas de tamanho 20. As Figuras 12 e 13 ilustram os resultados para temperaturas mínimas e máximas, respectivamente.

Quanto aos erros, obtivemos os valores de MAE=2.43 e MSE=8.85 (temperaturas mínimas), e MAE=2.16, MSE=6.87 (temperaturas máximas). Os erros são, conforme esperado, maiores que aqueles calculados sobre o conjunto de validação. Visualmente, também observamos piora considerável no resultado.

D. *Previsão de valores extremos*

Prever valores extremos é um desafio típico ao se trabalhar com séries temporais: por serem raros e aleatórios, é difícil ensiná-los aos modelos. No nosso caso, mesmo as redes com melhor desempenho apresentaram essa limitação, conforme ilustrado pelos gráficos das seções anteriores. Em um único experimento — com a rede LSTM básica para previsão de temperaturas máximas com janelas de tamanho 10 —, o modelo conseguiu prever um valor extremo sobre o conjunto de validação, o que podemos ver na Figura 14. O gráfico mostra, porém, que esse valor influenciou negativamente a previsão seguinte.

Após os experimentos de *baseline*, o grupo definiu, como objetivo para o trabalho final, a investigação sobre como melhorar a performance do modelo nesses casos. Embora não tenhamos implementado soluções nesse sentido, esta seção descreve sugestões da literatura para lidar com a questão.

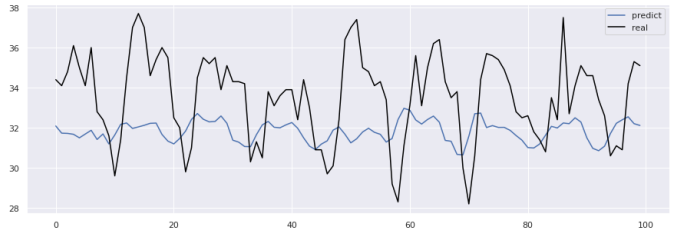

Figure 10. Dados reais e previsões de temperaturas máximas da Transformer com janelas de tamanho 10.

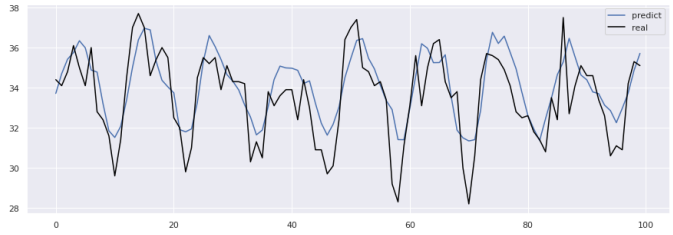

Figure 11. Dados reais e previsões de temperaturas máximas da Transformer com janelas de tamanho 20.

Ding et al. [15] exploram solução inspirada pela Teoria de Valores Extremos — *Extreme Value Theory* —, que passa pela criação de uma nova métrica de erro: *Extreme Value Loss*, ou EVL. A essência do trabalho é combinar o uso dessa métrica a um módulo de rede de memória adaptada, que consiga evitar tanto o *underfitting* quanto o *overfitting* comuns na previsão de extremos.

O *underfitting* acontece quando o modelo aprende adequadamente o padrão da série, mas não consegue prever os valores extremos — como o que ocorreu no nosso caso. O *overfitting*, ao contrário, é a adaptação excessiva do modelo a esses eventos irregulares, o que o torna bom para prevê-los, mas ruim para generalizar o restante da série.

A ideia dos autores, então, passa por: (a) usar uma métrica de erro mais adequada — afinal, o erro quadrático, métrica padrão desses modelos, penaliza a previsão de valores extremos desproporcionalmente em relação a valores regulares; (b) treinar uma rede capaz de memorizar eventos extremos, dado que, por vezes, eles exibem algum grau de regularidade temporal. Para tanto, são implementadas janelas históricas e um mecanismo de atenção na rede.

Laptev et al. [16], por sua vez, propõem uma abordagem de *auto feature extraction* prévia à alimentação de uma rede LSTM, combinada a um algoritmo de estimativa de incerteza. O modelo apresentou resultados satisfatórios sobre os dados dos autores, cujo principal objetivo era prever anomalias nas corridas do aplicativo Uber. Além de dados do Uber em si, foram usados dados climáticos e informações de calendário.

Não há maneira única de lidar com o problema: as sugestões que apresentamos são algumas dentre as muitas disponíveis na literatura. Em geral, propostas bem sucedidas combinam elementos distintos — dentre arquitetura de rede(s), variedade de dados e abordagens matemáticas.

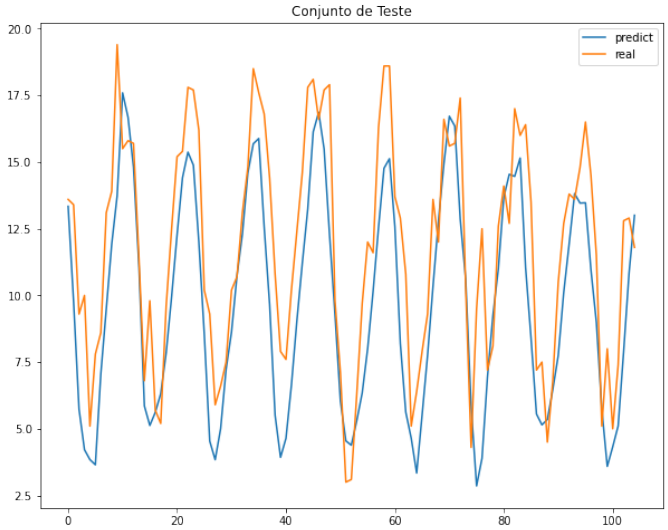

Figure 12. Resultado da melhor rede sobre conjunto de teste (temperaturas mínimas).

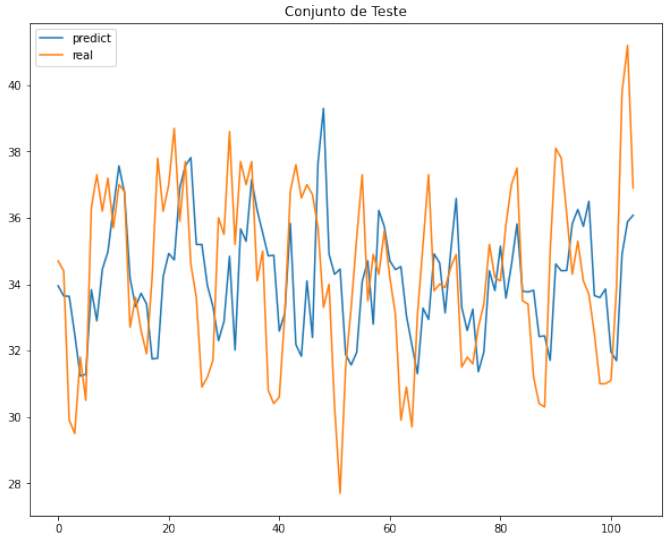

Figure 13. Resultado da melhor rede sobre conjunto de teste (temperaturas máximas).

IV. Considerações finais

Os resultados mostram que as redes escolhidas são adequadas para realizar a tarefa de previsão em séries temporais de dados climáticos. Embora a LSTM bidirecional tenha apresentado o melhor resultado, acreditamos que arquiteturas baseadas em Transformer são promissoras neste domínio e que, com maiores ajustes, poderia atingir resultados ainda melhores. De modo geral, o uso da recorrência parece ser uma abordagem acertada.

As limitações deste trabalho incluem — além da já mencionada dificuldade de previsão de valores extremos — o baixo volume de dados e a necessidade de executar os modelos separadamente para cada variável. Trata-se, porém, de limitações superáveis, que podem ser exploradas em trabalhos futuros.

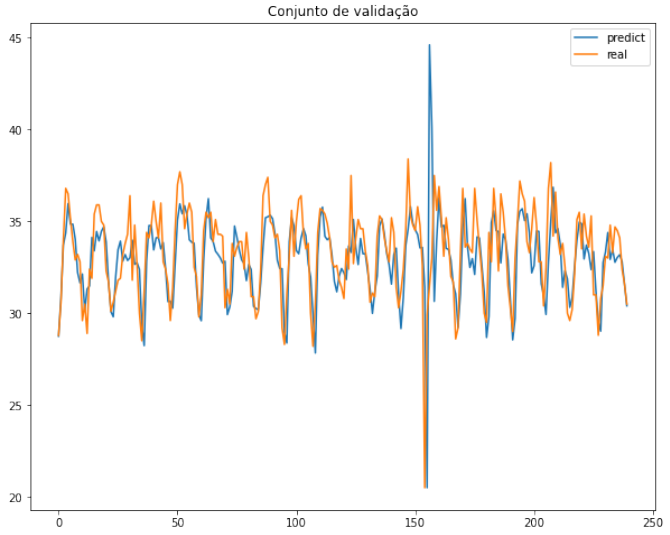

Figure 14. Valores reais e previsões de temperaturas máximas da rede LSTM básica com janelas de tamanho 10.

O grupo acredita que o objetivo inicial foi alcançado. O trabalho proporcionou aprendizados técnicos relevantes sobre o uso de deep learning em dados tabulares e em previsões de séries temporais, indicando um caminho interessante de exploração.

A. *Trabalhos futuros*

Trabalhos futuros que lidem com a questão que apresentamos podem explorar:

- Implementação de outras arquiteturas de rede;
- Aprimoramento da rede baseada em Transformer;
- Realização de experimentos com outros dados climáticos, de outras fontes e/ou com mais instâncias — por exemplo, dados diários;
- Previsão de janelas maiores de valores de saída, de modo a explorar tendências climáticas futuras;
- Implementação de soluções que corrijam a previsão de valores extremos.

Referências

[1] Mikael Huss. *Tabular Data and Deep Learning: Where Do We Stand?* http://shorturl.at/fvzH7. 2020.

[2] Sercan O. Arik and Tomas Pfister. "TabNet: Attentive Interpretable Tabular Learning". In: *arXiv* (2020).

[3] Dian Maharani, Hendri Murfi, and Yudi Satria. "Performance of Deep Neural Network for Tabular Data - A Case Study of Loss Cost Prediction in Fire Insurance". In: *International Journal of Machine Learning and Computing* 9.6 (2019), pp. 734–742.

[4] Lei Xu et al. "Modeling Tabular Data using Conditional GAN". In: *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)* (2019).

[5] Malay Haldar et al. "Applying Deep Learning To Airbnb Search". In: *arXiv* (2018).

[6] Rachel Thomas. *An Introduction to Deep Learning for Tabular Data*. http://shorturl.at/tLOTV. 2018.

[7] Meng-Hua Yen et al. "Application of the deep learning for the prediction of rainfall in Southern Taiwan". In: *Scientific Reports* 9 (2019).

[8] Tony Zhou. *Deep Learning for Time Series and why DEEP LEARNING?* http://shorturl.at/boxS1. 2020.

[9] Nenad Tomašev et al. "AI for social good: unlocking the opportunity for positive impact". In: *Nature Communications* 11 (2020).

[10] Departamento de Engenharia de Biossistemas da Escola Superior de Agricultura "Luiz de Queiroz" (ESALQ), Universidade de São Paulo. *Série de Dados Climatológicos do Campus Luiz de Queiroz de Piracicaba, SP*. http://www.leb.esalq.usp.br/leb/postocon.html. 2020.

[11] Jason Brownlee. *How to Develop LSTM Models for Time Series Forecasting*. https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/. 2020.

[12] Ashish Vaswani et al. *Attention Is All You Need*. https://arxiv.org/abs/1706.03762. 2017.

[13] Isaac Godfried. *Attention for time series forecasting and classification*. https://towardsdatascience.com/attention-for-time-series-classification-and-forecasting-261723e0006d. 2019.

[14] Husein Zolkepli. *Attention is all you need for stock forecasting*. https://github.com/huseinzol05/Stock-Prediction-Models/blob/master/deep-learning/16.attention-is-all-you-need.ipynb. 2019.

[15] Daizong Ding et al. "Modeling Extreme Events in Time Series Prediction". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery Data Mining*. KDD '19. Anchorage, AK, USA: Association for Computing Machinery, 2019, pp. 1114–1122. ISBN: 9781450362016. DOI: 10.1145/3292500.3330896.

[16] Nikolay Laptev et al. "Time-series Extreme Event Forecasting with Neural Networks at Uber". In: *ICML 2017 Time Series Workshop*. Sydney, Australia, 2017. URL: http://www.cs.columbia.edu/~lierranli/publications/TSW2017_paper.pdf.