Deep Learning sobre dados meteorológicos tabulares: Baseline

Matheus Ataide, 147375 | Mauricio Araujo, 184477 | Raysa Benatti, 176483

1 Introdução

O presente trabalho tem como proposta aplicar técnicas de deep learning para previsão em séries temporais de dados meteorológicos tabulares (estruturados). Nesta etapa, realizamos as seguintes tarefas:

- Troca da base de dados definida na etapa anterior e coleta de nova base de dados, conforme explicado na Seção 2;
- Pré-processamento dos dados, conforme explicado na Seção 2.1;
- Implementação de um modelo de rede neural recorrente LSTM (*Long Short-Term Memory*) básico sobre os dados, conforme explicado na Seção 3;
- Análise dos resultados, descritos na Seção 4;
- Discussão sobre o experimento e definição de incrementos a serem feitos para a versão final do proejeto, conforme explicado na Seção 5.

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

2 Base de Dados

Originalmente, o objetivo do grupo era trabalhar com a base de dados climáticos (série histórica) disponibilizada pelo INMET (Instituto Nacional de Meteorologia) via requisição. Havíamos coletado arquivos referentes a três estações meteorológicas da cidade de São Paulo (Horto Florestal, Mir. de Santana e Iag), o que totalizava aproximadamente 150 mil instâncias. Além disso, pretendíamos gerar predições de sete diferentes *features*, descritas na proposta inicial.

A exploração desses dados, contudo, revelou dificuldades: muitas das instâncias dessa base estão preenchidas parcial ou totalmente com NaN (*not a number*), o que inviabilizou a realização do experimento. Contudo, decididos a seguir com o objetivo de trabalhar no domínio ambiental e explorar ferramentas de deep learning sobre dados tabulares, optamos por usar outra base de dados meteorológicos.

Coletamos, então, o arquivo de valores absolutos de métricas climáticas mensais da estação convencional do Posto Meteorológico de Piracicaba, SP [1]. O arquivo traz, para todos os meses desde 1917, os valores de onze *features* medidas na estação: temperaturas máxima e mínima, chuvas diárias máxima e mínima, número de dias com chuva, umidades relativas máxima e mínima, velocidades instantâneas diárias máxima e mínima, velocidade média instantânea mensal e direção predominante do vento. Para máximas

(mínimas), o registro considera o maior (menor) valor observado no mês. Optamos por trabalhar somente com os valores de temperaturas máximas e mínimas, por três motivos: relevância da métrica em análises climáticas; suficiência para testar e avaliar nosso modelo; constância de observação, com baixa quantidade de entradas vazias nessa base — ao contrário de outras métricas cujas observações começaram a ser registradas mais tardiamente.

O arquivo tem 1235 instâncias: uma para cada um dos doze meses, durante 103 anos, exceto para 2020 (que tem registros até novembro). Apesar de muito menor que a base do INMET, o uso do conjunto de dados de Piracicaba se mostrou mais adequado devido a sua consistência, tendo poucas entradas vazias — o que refletiu nos resultados que descreveremos adiante.

2.1 Pré-processamento

A aplicação do modelo escolhido para previsão em séries temporais exige que os dados estejam ordenados cronologicamente. Para tanto, foi necessário pré-processar os dados coletados, organizando-os de maneira a exibir registros antigos sempre antes de registros novos. Além disso, o pré-processamento incluiu renomear colunas, converter tipos e deletar instâncias com valores NaN, para viabilizar as operações.

3 Metodologia

A implementação da rede LSTM seguiu a sugestão elaborada por Jason Brownlee [2]. O autor descreve sete modelos LSTM para previsão em séries temporais, cinco dos quais univariados (*vanilla*, *stacked*, bidirecional, CNN e ConvLSTM) e dois multivariados (*Multiple Input Series* e *Multiple Parallel Series*). Nesta etapa, o objetivo foi investigar a viabilidade de aplicar esse tipo de método sobre a base que coletamos. Assim, embora tenhamos selecionado duas variáveis de interesse (temperatura máxima e temperatura mínima), o experimento de baseline foi realizado somente com uma delas (temperatura máxima). Implementamos, então, o modelo de LSTM *vanilla*, ou básico.

A aplicação do modelo consiste em:

- Dividir os dados em três conjuntos para treinamento, validação e teste — escolhemos a proporção 70:20:10;
- Definir "janelas"de instâncias, com tamanho customizável, representado pela variável `n_steps` (no nosso experimento, usamos janelas de 10 valores de entrada para cada saída);
- Definir a quantidade de *units* de rede, a função de ativação, o otimizador, a métrica de erro e a quantidade de épocas.

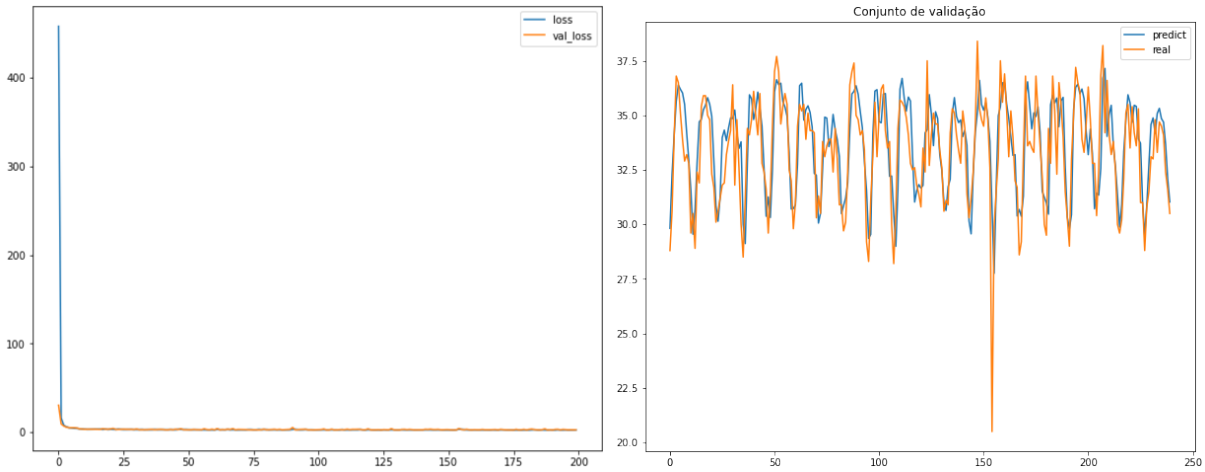
No nosso experimento, rodamos duas versões do modelo, variando a quantidade de *units*. Os demais parâmetros foram mantidos: ReLU como função de ativação, Adam como otimizador, erro quadrático médio (MSE) como métrica de erro e 200 épocas. A arquitetura de ambas as versões da rede consiste em duas camadas, sendo a primeira conectando a entrada com as unidades LSTM, e a segunda conectando estas unidades com um neurônio de saída. As duas versões foram treinadas sobre os dados de treino e validadas — com plotagem da evolução de *loss* e previsões — sobre o conjunto de validação. Por fim, rodamos e avaliamos a melhor das versões sobre o conjunto de teste.

4 Resultados

4.1 Versão 1: 50 *units*

A Figura 1 resume os resultados da Versão 1.

Figura 1: À esquerda, evolução da *loss* para o conjunto de treino (azul) e de validação (laranja). À direita, comparação entre as predições reais (laranja) e do modelo (Versão 1) sobre o conjunto de validação (azul).



4.2 Versão 2: 15 *units*

A Figura 2 resume os resultados da Versão 2. A Versão 2 mostrou valores de *loss* mais consistentes entre os conjuntos de treino e de validação; em razão disso, a aplicamos também sobre o conjunto de teste. A Figura 3 compara as previsões da Versão 2 sobre o conjunto de teste e as previsões reais.

Figura 2: À esquerda, evolução da *loss* para o conjunto de treino (azul) e de validação (laranja). À direita, comparação entre as predições reais (laranja) e do modelo (Versão 2) sobre o conjunto de validação (azul).

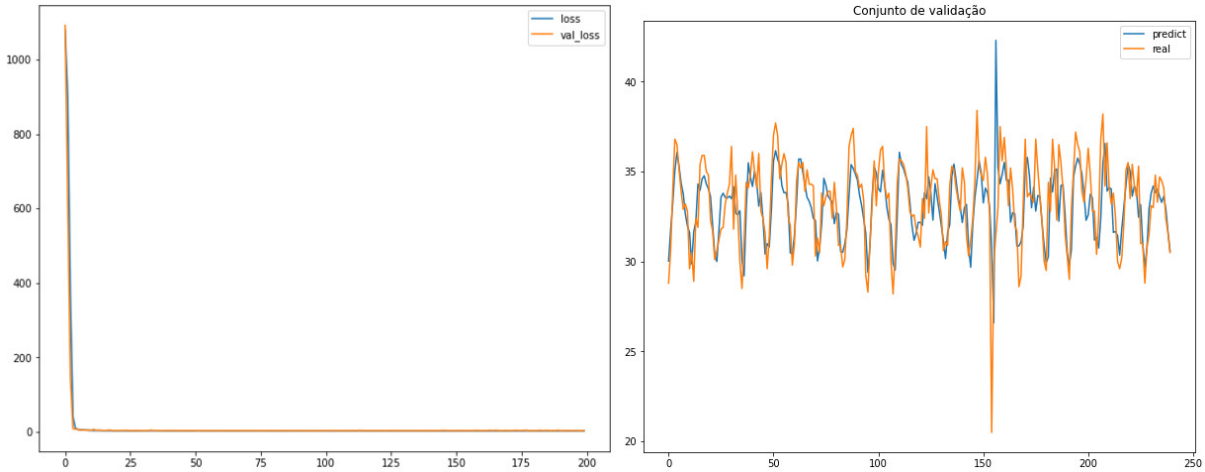
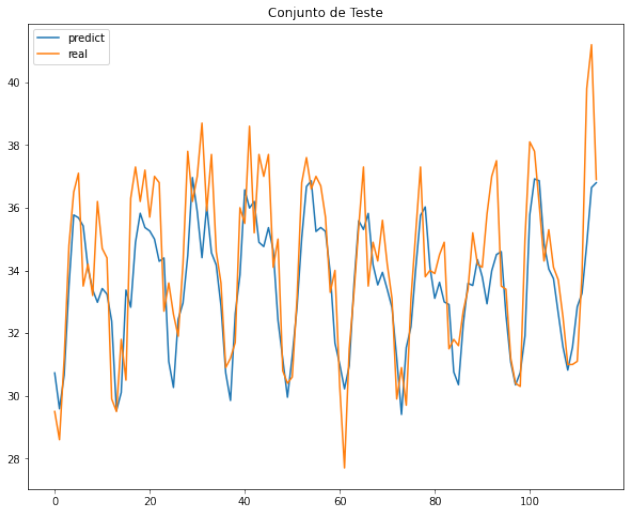


Figura 3: Comparação da Versão 2 entre as predições reais (laranja) e do modelo sobre o conjunto de teste (azul).



5 Discussão e próximos passos

Segundo nosso experimento, usar menos *units* — 15 em vez de 50 — levou a valores de *loss* mais consistentes. Notamos, contudo, que as duas versões do modelo performaram satisfatoriamente, o que indica a adequação dessa abordagem para realizar previsões em séries temporais. Prever valores extremos parece ser uma dificuldade do modelo.

Para a versão final do projeto, possíveis incrementos incluem: testar diferentes tamanhos de janela; explorar outras arquiteturas LSTM sobre nosso problema; implementar ao menos um modelo LSTM multivariado para previsão em séries temporais — efetuando, assim, previsões sobre nossas duas variáveis de interesse; investigar como melhorar a performance da previsão de valores extremos; avaliar a viabilidade de implementar um modelo de previsão em séries temporais baseado em Transformers.

Referências

[1] 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.

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