

Viabilidade do Uso de Inteligência Artificial para Simular Modelos Numéricos que Estimam o Acúmulo de Precipitação Através da Microfísica de Nuvens

Gabrielly Vieira Leal de Sena
Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo
São José dos Campos, SP
gabrielly.sena@unifesp.br

Livia Beatriz Alves Canuto Ferreira de Paula
Instituto de Ciência e Tecnologia
Universidade Federal de São Paulo
São José dos Campos, SP
livia.canuto@unifesp.br

1. Introdução e Motivação

Atualmente, as medições e previsões de precipitação são realizadas utilizando modelos matemáticos complexos, conhecidos como modelos numéricos de previsão do tempo. Esses modelos dividem a atmosfera em grades tridimensionais e aplicam equações físicas para simular seu comportamento ao longo do tempo.

As saídas desses modelos são interpretadas por meteorologistas e profissionais especializados para produzir previsões meteorológicas finais. Isso envolve uma análise cuidadosa das condições atmosféricas simuladas para estimar a ocorrência e intensidade da precipitação (Sampaio; Dias, 2014).

A execução dos modelos numéricos demanda um tempo significativo devido à complexidade das equações e à necessidade de assimilação de grandes volumes de dados observacionais, exigindo alto poder de processamento para executar simulações detalhadas e precisas.

A Inteligência Artificial (IA) pode potencialmente otimizar o processo de simulação, reduzindo a dependência de capacidade computacional massiva e acelerando a produção de previsões precisas e confiáveis. Além disso, considerando as mudanças climáticas, os padrões de precipitação estão se tornando mais variáveis e difíceis de prever com modelos tradicionais - o uso de IA pode auxiliar na adaptação a essas mudanças.

2. Conceitos Fundamentais

No processo de previsão do tempo, são utilizadas diversas medidas extraídas da atmosfera para calcular resultados específicos. Neste trabalho, o objetivo é estimar a Taxa de Precipitação Acumulada (ACPPRR, do inglês Accumulated Precipitation Rate) no intervalo de 3600 segundos (1 hora), para cada par ordenado (x, y) . Utilizamos um conjunto de 13 variáveis de entrada, detalhadas a seguir.

Temos 10 variáveis formadas por arrays tridimensionais, contendo valores aferidos para cada (x, y, z) :

- **THP (Produção Total de Calor):** Medida do calor total produzido na atmosfera, indicativa da intensidade dos processos de aquecimento.

- **Theta (Potencial Termodinâmico):** Indica a estabilidade ou instabilidade da atmosfera com base nas diferenças verticais de temperatura.
- **PP (Precipitação):** Quantidade de precipitação observada ou prevista em determinado período.
- **RTP (Potencial de Transporte Relativo):** Capacidade da atmosfera de transportar umidade em relação às condições de vento.
- **RV (Vorticidade Relativa):** Medida da rotação do ar em relação a uma região específica, indicando a tendência à formação de sistemas atmosféricos.
- **WP (Potência do Vento):** Energia cinética associada ao movimento do vento, influenciando os padrões atmosféricos locais.
- **DN0 (Deformação N0):** Taxa de variação da forma e orientação das partículas de ar em nível inicial.
- **PI0 (Instabilidade Potencial):** Potencial de instabilidade atmosférica, relacionado à tendência de desenvolvimento de nuvens convectivas.
- **RCP (Potencial Convectivo Relativo):** Indicação da probabilidade de ocorrência de convecção atmosférica em relação ao ambiente circundante.
- **RRP (Pico de Taxa de Chuva):** Máximo esperado da taxa de precipitação durante um evento de chuva intensa.

E outras 3 variáveis formadas por arrays bidimensionais (x, y) :

- **RTGT (Tendência à Direita):** Tendência da trajetória de um ciclone ou anticiclone para a direita, devido ao efeito da rotação da Terra.
- **ACPPRR (Taxa de Precipitação Acumulada):** Total acumulado de precipitação em um determinado intervalo de tempo.
- **PCPPRR (Pico de Taxa de Precipitação em Chuva):** Taxa máxima de precipitação por unidade de tempo durante um evento de chuva.

Neste estudo, dispomos de dois conjuntos de dados: um maior, que denominaremos de big data, e outro menor, denominado tiny data. O conjunto tiny data possui dimensões de $31 \times 38 \times 41$ (X, Y, Z) , enquanto o big data apresenta dimensões de $540 \times 540 \times 49$ (X, Y, Z) . Essas matrizes

tridimensionais representam uma porção da atmosfera da América do Sul.

Além disso, na parte prática do desenvolvimento usaremos conceitos técnicos tais como:

- **Análise Exploratória:** A análise exploratória de dados é uma abordagem que envolve técnicas e métodos para explorar e sumarizar características principais de um conjunto de dados. Seu objetivo principal é extrair insights preliminares, identificar padrões, detectar anomalias e formular hipóteses que possam orientar análises mais aprofundadas e a tomada de decisões informadas.
- **Inteligência Artificial:** Inteligência artificial (IA) refere-se à capacidade de sistemas computacionais realizarem tarefas que normalmente exigiriam inteligência humana. Isso inclui a capacidade de aprender com experiência passada, ajustar-se a novas entradas e realizar tarefas complexas como reconhecimento de padrões, processamento de linguagem natural, tomada de decisões e resolução de problemas.
- **Aprendizado de Máquina:** Campo da inteligência artificial que se concentra no desenvolvimento de algoritmos e modelos que permitem aos computadores aprenderem a partir de dados e tomar decisões automáticas sem serem explicitamente programados para cada tarefa específica. O cerne do aprendizado de máquina está na capacidade dos algoritmos de identificar padrões nos dados e utilizar esses padrões para fazer previsões ou tomar decisões.
- **Regressão:** Problema de aprendizado de máquina onde o objetivo é prever um valor contínuo para uma variável dependente com base em uma ou mais variáveis independentes.
- **Redes Neurais Artificiais:** Modelos computacionais inspirados pelo funcionamento do cérebro humano, compostos por unidades interconectadas chamadas de neurônios artificiais. Essas redes são capazes de aprender padrões complexos nos dados através de um processo de treinamento supervisionado, onde ajustam os pesos das conexões entre os neurônios para otimizar a capacidade de realizar tarefas específicas, como classificação ou previsão.

3. Trabalhos Relacionados

- **Artificial neural network technique for rainfall forecasting applied to the São Paulo region.**
 - O artigo apresenta um estudo sobre a aplicação de redes neurais artificiais para a previsão de precipitação e conclui o modelo RNA como promissor. Por fim faz uma comparação entre RNA e modelos de regressão linear revelando que redes neurais artificiais podem trazer resultados superiores.
- **Can machines learn to predict weather? using deep learning to predict gridded 500-hpa geopotential height from historical weather data.**
 - O estudo faz uso de redes neurais convolucionais

(CNNs), sem conhecimento dos processos físicos, para previsão do tempo. O modelo obteve resultados interessantes apesar de não ter superado um modelo totalmente físico.

- **Predição de níveis de precipitação em Catalão-Goiás com base em séries temporais de múltiplos pontos geográficos.**
 - Este artigo faz uso de uma rede neural chamada long short-term memory (LSTM) para fazer previsão de chuva. No artigo foi considerado tanto tarefas de regressão quanto de classificação. Propõe que a diferença entre o período de tempo (diária ou horária) tem impacto nos resultados.
- **Uma avaliação de algoritmos de regressão para predição de volume de chuva.**
 - Faz uma comparação de desempenho entre diferentes algoritmos de regressão para previsão de volume de chuva a cada hora. Mostra que os algoritmos cujos algoritmos possuem os menores erros médios e cujos poderiam ser estudados para previsão de chuva em estudos futuros.
- **Deep learning models for the prediction of rainfall.**
 - O artigo busca determinar o menor erro entre os modelos de redes neurais de memória de curto-longo prazo (LSTM) e rede neural convolucional (CNN).
- **Evolução dos Modelos Climáticos e de Previsão de Tempo e Clima.**
 - O estudo se propõe a explicar como foi a evolução da previsão de tempo, como é feita hoje e quais são as limitações encontradas.
- **Rainfall estimation using artificial neural network group. Neurocomputing.**
 - Esse artigo utiliza redes neurais para estimar precipitação. No estudo, os erros médios foram reduzidos e os cálculos de precipitação também tiveram o tempo reduzido.

4. Objetivo

O objetivo deste trabalho é explorar a viabilidade do uso de Inteligência Artificial para simular modelos numéricos que estimam precipitação por meio da microfísica das nuvens. A proposta envolve estimar a taxa de acumulação de precipitação para cada ponto dentro de um espaço definido, utilizando variáveis iniciais específicas, conforme citado em "Conceitos Fundamentais". A principal abordagem consiste em comparar os resultados obtidos pela Inteligência Artificial com os do modelo numérico tradicional, buscando verificar a precisão e eficácia da metodologia.

5. Metodologia Experimental

A realização do protocolo experimental completo foi testada inicialmente no menor conjunto de dados devido à agilidade, onde foi possível avaliar a compatibilidade entre

modelos e formato dos dados. Posteriormente aplicada ao maior, contendo uma amostra mais ampla e portanto sendo o foco no desenvolvimento.

O treinamento se deu no Google Colab e a linguagem utilizada foi Python, devido a suas bibliotecas que auxiliam na aprendizagem de máquina.

No que se refere ao protocolo, a base de dados precisou ser formatada para que as informações pudessem ser passíveis da realização de análise exploratória - Dataframe da biblioteca Pandas, para isso foram necessárias bibliotecas adicionais, como Re e NumPy.

Dessa maneira, como as entradas possuem dimensões distintas (2D e 3D), detém-se dois arquivos de entrada e um arquivo de saída:

df_input1

	X	Y	Z	thp	theta	pp	rtp	rv	vp	dn0	pi0	rcp	rrp
0	1	1	1	282.160370	282.167267	0.014511	5.242808e-03	5.253425e-03	0.000000	1.187878	1009.854450	0.0	0.0
1	1	1	2	282.163513	282.167297	-0.093423	5.241041e-03	5.253425e-03	0.006589	1.181401	1007.646730	0.0	0.0
2	1	1	3	282.282257	282.284576	-0.210717	4.487439e-03	4.488973e-03	0.009075	1.174365	1005.240230	0.0	0.0
3	1	1	4	282.306030	282.305817	-0.339313	4.438843e-03	4.441196e-03	0.009469	1.166724	1002.617190	0.0	0.0
4	1	1	5	282.298462	282.298035	-0.479053	4.458640e-03	4.461752e-03	0.009610	1.158430	999.758057	0.0	0.0
...
14871595	540	540	47	504.486694	504.486536	-0.236961	3.170561e-07	3.170223e-07	0.030950	0.079090	421.280548	0.0	0.0
14871596	540	540	48	522.630249	522.617798	-0.201759	1.000000e-08	1.000001e-08	0.017074	0.070003	406.933350	0.0	0.0
14871597	540	540	49	541.134644	541.124268	-0.173992	1.000000e-08	1.000000e-08	0.006089	0.061962	393.086639	0.0	0.0
14871598	540	540	50	560.339233	560.329651	-0.149532	1.000000e-08	1.000000e-08	0.000000	0.054904	379.716125	0.0	0.0
14871599	540	540	51	579.552979	579.536034	-0.125073	1.000000e-08	1.000000e-08	0.000000	0.048688	366.793365	0.0	0.0

14871600 rows x 13 columns

(a) Tabela *df_input1*

df_input2

	X	Y	rtgt	accpr	pcpr
0	1	1	1.0	0.0	0.0
1	1	2	1.0	0.0	0.0
2	1	3	1.0	0.0	0.0
3	1	4	1.0	0.0	0.0
4	1	5	1.0	0.0	0.0
...
291595	540	536	1.0	0.0	0.0
291596	540	537	1.0	0.0	0.0
291597	540	538	1.0	0.0	0.0
291598	540	539	1.0	0.0	0.0
291599	540	540	1.0	0.0	0.0

291600 rows x 5 columns

(b) Tabela *df_input2*

df_output

	accpr_output
0	0.0
1	0.0
2	0.0
3	0.0
4	0.0
...	...
291595	0.0
291596	0.0
291597	0.0
291598	0.0
291599	0.0

291600 rows x 1 columns

(c) Tabela *df_output*

Figura 1: Formato das tabelas

Com os dados necessários devidamente formatados, suas características foram exploradas para compreensão do comportamento de maneira eficaz.

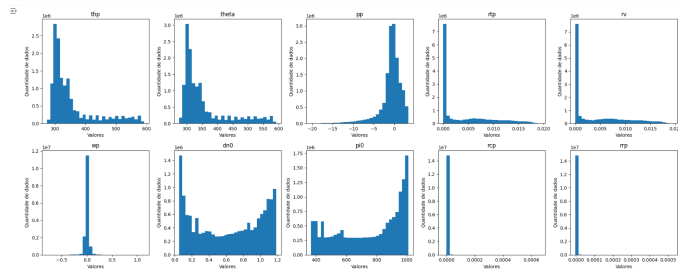


Figura 2: Distribuição de *df1*

Na distribuição dos dados de *df1*, observa-se que as variáveis *rtp*, *vp*, *dn0* e *rcp* possuem valores muito próximos de zero. As variáveis *thp* e *theta* apresentam uma distribuição semelhante, com valores que variam aproximadamente de 270 a 600, enquanto *pi0* varia de 370 a 1000. Por outro lado, *pp* possui valores negativos, variando de -17 até 3,5. Devido a essa discrepância nos valores, será necessário normalizar os dados. Além disso, nota-se que as variáveis não seguem uma distribuição normal.

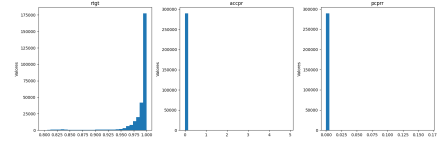


Figura 3: Distribuição de *df2*

No conjunto de dados *df2*, nota-se que a distribuição dos dados para *accpr* e *pcpr* se concentra bastante próxima de zero, enquanto para *rtpgt*, os valores variam entre 0.8 e 1. Além disso, os dados também não seguem uma distribuição normal.

No *dfout*, é perceptível que os dados são muito pequenos e se concentram perto de zero, sem valores negativos. Isso faz sentido considerando que estamos analisando o acúmulo de precipitação.

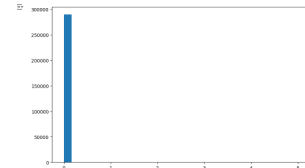


Figura 4: Distribuição de *df_output*

Na etapa de preparação dos dados, além da investigação sobre os tipos de dados, a presença de dados ausentes e registros duplicados, foram identificadas duas necessidades principais: normalização e avaliação de correlação.

Para a normalização, as colunas 'X', 'Y' e 'Z' foram descartadas e aplicado o método MinMaxScaler da biblioteca Scikit-learn, que transforma os valores para o intervalo entre 0 e 1.

df_input1 normalizada

	thp	theta	pp	rtp	rv	vp	dn0	pi0	rcp	rrp
0	0.026324	0.023458	0.855477	0.270398	2.706556e-01	0.411164	9.999997e-01	9.999998e-01	0.034327	0.008736
1	0.026334	0.023458	0.850931	0.270307	2.706556e-01	0.414569	9.943136e-01	9.965666e-01	0.034327	0.008736
2	0.026706	0.023820	0.845976	0.231502	2.312712e-01	0.415854	9.881373e-01	9.920244e-01	0.034327	0.008736
3	0.026781	0.023895	0.840575	0.229000	2.288097e-01	0.416058	9.814303e-01	9.887454e-01	0.034327	0.008736
4	0.026757	0.023870	0.834690	0.230020	2.288680e-01	0.416131	9.741495e-01	9.842992e-01	0.034327	0.008736
...
14871595	0.724443	0.723575	0.844886	0.000450	1.633289e-05	0.427160	2.668702e-02	6.473098e-02	0.034327	0.008736
14871596	0.781415	0.780735	0.846368	0.000434	5.151887e-07	0.419988	1.871115e-02	6.242820e-02	0.034327	0.008736
14871597	0.839620	0.839019	0.847538	0.000434	5.151470e-07	0.414311	1.165259e-02	4.088771e-02	0.034327	0.008736
14871598	0.899624	0.899605	0.848568	0.000434	5.151470e-07	0.411164	5.456086e-03	2.009574e-02	0.034327	0.008736
14871599	0.960156	0.959990	0.849598	0.000434	5.151470e-07	0.411164	9.814510e-01	4.665185e-08	0.034327	0.008736

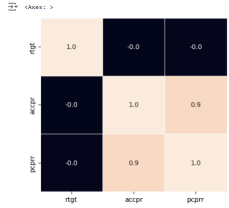
14871600 rows x 10 columns

Figura 5: Tabela *df_input1* normalizada

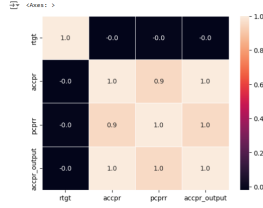
Quanto à avaliação de correlação, cada um dos dois DataFrames de entrada foi tratado individualmente, além de integrar o DataFrame de saída ao DataFrame de entrada 2D.



(a) Correlação df_input1



(b) Correlação df_input2



(c) Correlação df_output

Figura 6: Correlação dos dataframes

Possuindo três DataFrames com dimensões distintas, foi necessário converter seus valores para referências unicamente de X e Y. Após essa transformação, consolidamos todos os dados em um único DataFrame. O resultado apresentou 1.178 linhas e 434 colunas na versão Tiny Data e 291.600 linhas e 514 colunas na versão Big Data.

Como o objetivo trata-se de estimar um valor contínuo que represente a quantidade de chuva acumulada, uma tarefa de regressão deve ser realizada. Na abordagem do problema foram utilizados três Modelos de Inteligência Artificial diferentes, um deles sendo uma Rede Neural. Além disso, todos obtiveram registros de tempo de execução em segundos, para treino e predição, através da biblioteca Time.

Para o treinamento, os dados foram divididos em conjuntos de treino e teste na proporção de 80 para treino e 20 para teste.

O algoritmo KNeighborsRegressor (KNR) foi escolhido para ser um dos modelos analisados, utilizando a biblioteca Scikit-Learn. Todos os parâmetros foram mantidos em seus valores padrão, exceto o `n_neighbors`, que foi definido. Esse parâmetro define a quantidade de vizinhos que serão considerados para fazer a predição. Inicialmente, foi calculado como a raiz quadrada do número total de linhas do dataframe, mais um, resultando em 541. As comparações com valores maiores e menores mostraram que o desempenho piorava ao aumentar o valor, mas melhorava ao reduzi-lo ligeiramente. De forma arbitrária o valor 355 foi testado de forma experimental e mantido, buscando não reduzir mais o valor para evitar previsões instáveis.

Outro algoritmo utilizado para comparação foi o Support Vector Regression (SVR), também implementado com a biblioteca Scikit-Learn. Definimos o parâmetro kernel como linear, seguindo uma recomendação da própria documentação. A sugestão era adotar o kernel linear para datasets com mais de 10.000 amostras, devido ao tempo de processamento necessário para a predição.

A Rede Neural selecionada foi o algoritmo Multi-Layer Perceptron (MLP Regressor) da biblioteca Scikit-Learn. Ao utilizar MLP, diversos parâmetros foram configurados manualmente, enquanto os restantes foram mantidos nos valores padrão. Os escolhidos foram:

- **random_state:** Esse parâmetro controla a aleatoriedade envolvida no treinamento do modelo. Definimos um valor para garantir que os resultados seriam reproduzíveis. O valor escolhido foi 0.
- **tol:** Quando a perda não melhorar em pelo menos o valor de tol por 10 iterações, então o algoritmo vai parar o treinamento. Isso garante o early stop e ajuda a prevenir overfitting. O valor escolhido para o tol foi 1×10^{-9} .
- **max_iter:** Máximo de iterações que o modelo vai permitir que aconteça durante o treinamento. A escolha do valor foi 500.
- **activation:** É a função de ativação da camada oculta. A escolha foi feita com base em alguns testes. A que teve melhor score e menor erro foi escolhida. Nesse caso 'tanh'.

Além disso, para testar o desempenho da Rede Neural com diferentes amostras e diversificar a área de avaliação, foi utilizado um K-Fold de tamanho 5.

6. Resultados

As medidas de avaliação escolhidas foram Mean Squared Error (MSE) e Mean Absolute Error (MAE) tendo em vista que se trata de um problema de regressão. Além disso, para KNR e SVR, o Coeficiente de Determinação (R2 Score) também foi considerado, assim como MLPRegressor com o MLP Score.

Iniciando pelo KNR, os indicadores de desempenho seguindo as métricas escolhidas, apresentaram os seguintes resultados:

- **Tempo de Treino:** 0.41716 segundos.
- **Tempo de Predição:** 206.78257 segundos.
- **MSE:** 0.00007401
- **MAE:** 0.00045886
- **R2 Score:** 0.80253745.

O seu desempenho foi razoável, entretando seu tempo de predição é excessivamente demorado, por ser um modelo não-paramétrico. Seu processo de aprendizado a seguir.

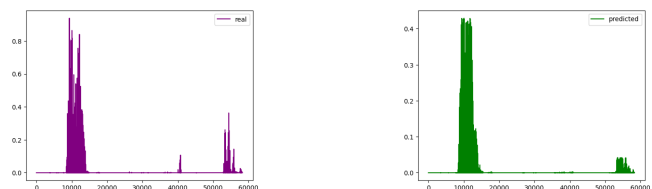


Figura 7: Gráficos do modelo KNR

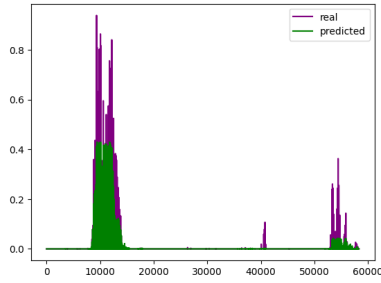


Figura 8: Predição e Real sobrepostos KNR

No que se refere ao SVR, os resultados foram bem menos satisfatórios, especialmente para R2 Score. Entretanto, seu tempo de execução é extremamente otimizado tanto para treino quanto para predição.

- **Tempo de Treino:** 3.05045 segundos.
- **Tempo de Predição:** 0.19387 segundos.
- **MSE:** 0.00035312.
- **MAE:** 0.01424280.
- **R2 Score:** 0.05786563.

Segue a comparação entre os dados reais e os previstos.

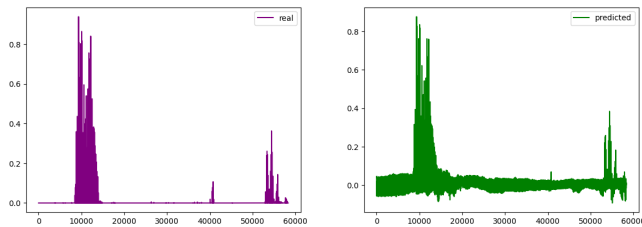


Figura 9: Gráficos do modelo SVR

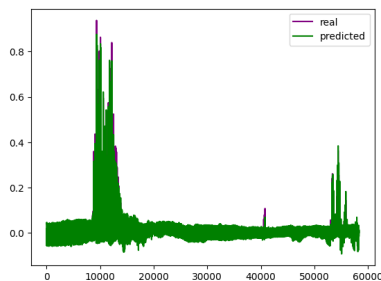


Figura 10: Predição e Real sobrepostos SVR

O modelo MLPRegressor se aproximou bastante dos valores reais, mesmo em casos de saltos ou quedas nos dados, demonstrando um desempenho satisfatório e alcançando ótimos resultados na base de dados. Seu desempenho em aprendizagem, conforme esperado, foi o melhor entre todos os modelos abordados.

- **Tempo de Treino:** 422.77400 segundos.
- **Tempo de Predição:** 0.30289 segundos.
- **MSE:** 0.00000296.

- **MAE:** 0.00010507.
- **MLP Score:** 0.9921146740729507.

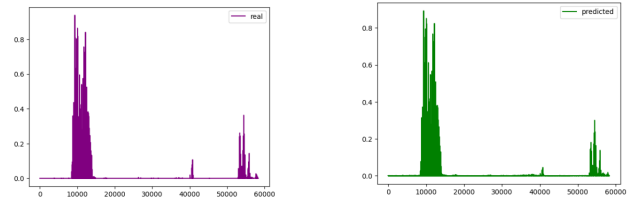


Figura 11: Gráficos do modelo MLP

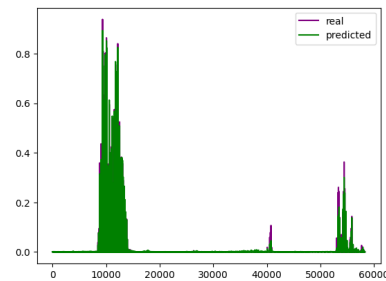


Figura 12: Predição e Real sobrepostos MLP

O gráfico de perda a seguir ilustra que a perda diminui ao longo das épocas, indicando que o modelo está se ajustando bem aos dados de treinamento.

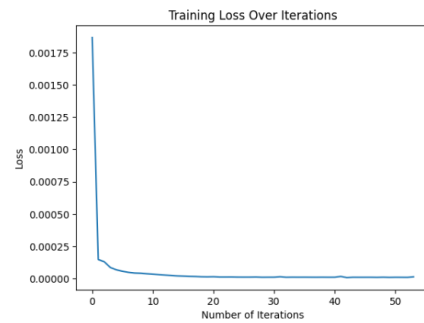
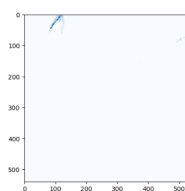
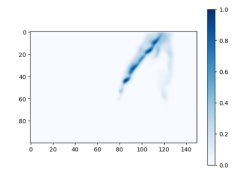


Figura 13: Gráfico de Perda

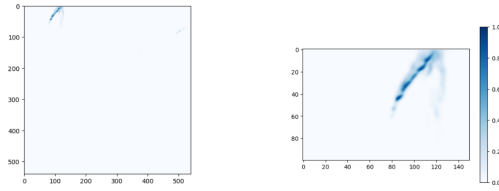
Além disso, com o objetivo de visualizar a distribuição original e a prevista pelo modelo no que se refere a concentração de chuva no plano (x,y), um cromograma foi adicionado.



(a) Ocorrência de chuva



(b) Ocorrência de chuva centralizada



(a) Previsão de ocorrência de chuva (b) Previsão de ocorrência de chuva centralizada

Ao rodar cada uma das 5 iterações do K-Fold os resultados foram majoritariamente similares ao treinamento original de MLPRegressor, entretanto, um fold demonstrou uma queda significativa de cerca de 0.08 pontos na avaliação score. O que demonstra que a maneira em que a divisão dos dados é realizada interfere diretamente no aprendizado e desempenho do modelo ainda que em uma visão geral seus resultados permaneçam positivos.

Iteração 1:

- **Tempo de Treino:** 115.10677 segundos.
- **Tempo de Predição:** 0.30792 segundos.
- **MSE:** 0.00003166.
- **MAE:** 0.00508331.
- **MLP Score:** 0.9129789997569485.

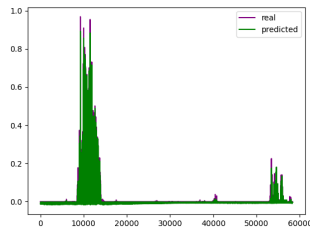


Figura 16: Gráfico predição e real. Iteração 1

Iteração 2:

- **Tempo de Treino:** 158.36279 segundos.
- **Tempo de Predição:** 0.29735 segundos.
- **MSE:** 0.00000436.
- **MAE:** 0.00127182.
- **MLP Score:** 0.9890824903991863.

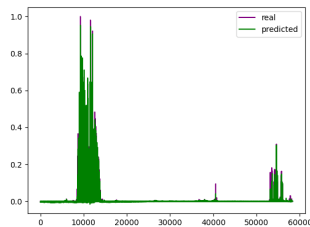


Figura 17: Gráfico predição e real. Iteração 2

Iteração 3:

- **Tempo de Treino:** 134.46335 segundos.
- **Tempo de Predição:** 0.30421 segundos.
- **MSE:** 0.00000232.
- **MAE:** 0.00045650.
- **MLP Score:** 0.9955942637299701.

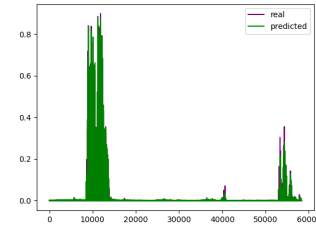


Figura 18: Gráfico predição e real. Iteração 3

Iteração 4:

- **Tempo de Treino:** 154.48142 segundos.
- **Tempo de Predição:** 0.29706 segundos.
- **MSE:** 0.00000543.
- **MAE:** 0.00184697.
- **MLP Score:** 0.9897884313843067.

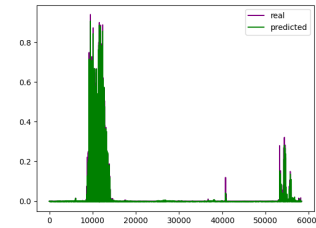


Figura 19: Gráfico predição e real. Iteração 4

Iteração 5:

- **Tempo de Treino:** 125.00992 segundos.
- **Tempo de Predição:** 0.30348 segundos.
- **MSE:** 0.00000326.
- **MAE:** 0.00109646.
- **MLP Score:** 0.9913112201398983.

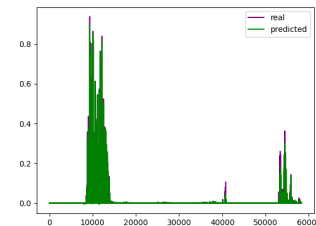


Figura 20: Gráfico predição e real. Iteração 5

Para todo o protocolo experimental, o ambiente utilizado foi o Google Colab. Entretanto, para uma estimativa mais precisa referente ao custo computacional, utilizou-se um computador com as seguintes especificações:

- AMD Ryzen 5 3400G.
- Frequência: 3,70 GHz.
- 4 núcleos.
- 8 processadores lógicos.
- Cache L1 384 KB.
- Cache L2 2,0 mb.
- Cache L3 4,0 mb.
- Memória sendo 2 slots de 8GB totalizando 16GB com 13,9GB utilizável
- Frequência: 2400 MHz cada memória.

Os resultados foram obtidos em um ambiente de execução limitado e os experimentos realizados em Python, o qual sendo uma linguagem interpretada, conta com um custo computacional adicional. Mas, foi suficiente para demonstrar o potencial da área de pesquisa.

7. Conclusões

Nas considerações finais, confirma-se a viabilidade do uso de Inteligência Artificial para simulação de modelos numéricos que estimam o acúmulo de precipitação através da microfísica de nuvens considerando os dados como pontos independentes no espaço. Nessa tarefa, o modelo de rede neural MLPregressor apresenta melhores indicadores de desempenho no que se refere a precisão de modelagem do problema. Além disso, como o treinamento é realizado uma única vez, o tempo de predição pode ser considerado satisfatório considerando a potência computacional disponível durante o desenvolvimento.

No que se refere a perspectivas futuras, seria interessante considerar os dados como dependentes e levar em conta mais informações de localização geográfica, como o clima da região no período do ano em que a amostra foi coletada, pois podem causar variação de comportamento.

No geral, trata-se de uma área propícia para ser explorada em trabalhos futuros, podendo trazer mudanças significativas no âmbito meteorológico.

Referências

- [1] RAMIREZ, M.C.V., VELHO, H.C.; FERREIRA, N.J. Artificial neural network technique for rainfall forecasting applied to the São Paulo region. *Journal of Hydrology*, v.301, n.1-4, p. 146-162, jan. 2005.
- [2] ASWIN, S.; GEETHA, P.; VINAYAKUMAR, R. Deep learning models for the prediction of rainfall. 2018 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2018.
- [3] FERREIRA, G. S. E.; MEDEIROS, D. Uma avaliação de algoritmos de regressão para predição de volume de chuva. *Anais do XL Simpósio Brasileiro de Telecomunicações e Processamento de Sinais*. Sociedade Brasileira de Telecomunicações, 2022.
- [4] RESENDE, G. S. et al. Predição de níveis de precipitação em Catalão-Goiás com base em séries temporais de múltiplos pontos geográficos / Precipitation forecast in Catalão-Goiás based on time series of multiple geographic points. *Brazilian Journal of Development*, v. 7, n. 4, p. 34960–34978, 2021.
- [5] SAMPAIO, G.; DIAS, P. L. DA S. Evolução dos Modelos Climáticos e de Previsão de Tempo e Clima. *Revista USP*, n. 103, p. 41, 2014.
- [6] WEYN, J. A.; DURRAN, D. R.; CARUANA, R. Can machines learn to predict weather? Using deep learning to predict gridded 500-hPa geopotential height from historical weather data. *Journal of advances in modeling earth systems*, v. 11, n. 8, p. 2680–2693, 2019.
- [7] ZHANG, M.; FULCHER, J.; SCOFIELD, R. A. Rainfall estimation using artificial neural network group. *Neurocomputing*, v. 16, n. 2, p. 97–115, 1997.