

# Reconhecimento de padrões climáticos com redes neurais hipercomplexas

Thiago S. Silva, Mario S. M. Santana

<sup>1</sup>Departamento de Matemática Aplicada – Universidade estadual de Campinas (UNICAMP)

t251676@dac.unicamp.br;

m223615@dac.unicamp.br;

**Abstract.** *Trying to predict the weather has direct benefits for agriculture, population well-being, and the prediction of extreme events. Its natural difficulty, due to chaotic events, makes it costly and time-consuming to increase the resolution of the grids.*

*To generate grids for global models to regional models, interpolation is currently widely used. This ignores factors such as altitude, local humidity, and the period we are evaluating, among others.*

*When the grid spacing is too large, interpolation is not effective. Therefore, training a neural network with local parameters can enable us to accelerate the grid construction process.*

**Resumo.** *Tentar prever o clima tem ganhos diretos na agricultura, bem estar da população e predição de eventos extremos. Sua dificuldade natural devida aos eventos caóticos, faz com que seja custoso e demorado aumentar a resolução das malhas.*

*Para gerar malhas de modelos globais para modelos regionais, atualmente é usado largamente a interpolação. O que ignora fatores como altitude, umidade do local, período que estamos avaliando, entre outros.*

*Quando a espaçamento da malha é muito grande, a interpolação não é eficaz. Assim, treinar uma rede neural com os parâmetros locais pode nos possibilitar acelerar o processo de construção de malhas.*

## 1. Introdução

Modelos globais são usados para previsões de longo prazo, assim, são referência nos relatórios emitidos pelo Painel Intergovernamental sobre Mudanças Climáticas (IPCC) para avaliar mudanças necessárias para o controle dos impactos severos. Um desses modelos é o mais usado, MIROC, tem intervalos de 2,8° de latitude e longitude e 40 camadas de 3hPa sobre a superfície terrestre. Sobre o oceano essa configuração muda para 1° a partir de 63° N. [Eyring et al. 2016]

Em geral, modelos regionais utilizam os modelos globais como um suporte para a criação de malhas de latitude e longitude mais refinadas, visto

que a resolução das malhas globais são muito grandes para captar fenômenos climáticos de ordem local. Cabe ressaltar que, para refinar as malhas, é muito comum que se realize um ajuste por meio da interpolação; sendo que, dos diversos meios de interpolação possíveis, existe um erro contínuo por aproximação e desconsideração de algumas informações na construção dos valores, como altura e umidade.

Visando encontrar métodos alternativos e que tentem contornar os problemas enfrentados pela interpolação, neste projeto, uma rede neural hipercomplexa, de álgebra quaterniônica, será proposta como um método útil para a realização do refinamento das malhas climáticas.

## 2. Informações sobre os Dados

Os dados usados foram desenvolvidos por [Alexandre C. Xavier 2022] usando dados de estação de satélite e outras fontes para construir uma malha com distribuição contínua para todo o Brasil, com o período de 1961 até 2022.

Por questões computacionais e de tempo de processamento, durante o treinamento, foram selecionadas as latitudes e longitudes presentes na figura (1). Note que a região dessa figura, corresponde a um quadrado, centrado no ponto de latitude -22.85 e longitude -47.05, com comprimento de lado de 6 pontos acima, abaixo, à direita e à esquerda de seu ponto central. **OBS.:** A malha de latitudes e longitudes considerada neste projetos engloba e está centrada na cidade de Campinas.

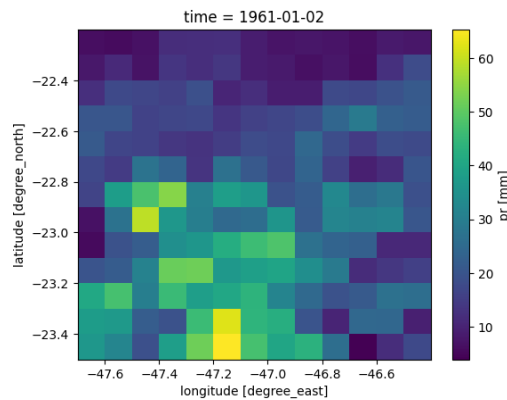
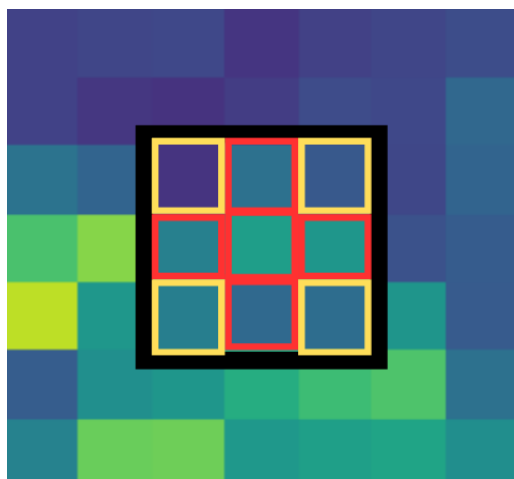


Figure 1. Região analisada no dia 2/1/2961

Em relação à estratégia de treinamento da rede neural, foi pensado o seguinte: para cada quatro pontos separados por um ponto, formando um quadrado, as extremidades são usadas como a entrada da rede, enquanto o interior foi usado como os gabaritos para as previsões da rede. Esses fatos serão melhor explicados na próxima seção, mas para já ir se acostumando, veja a imagem (2) para uma melhor compreensão do que foi dito. Ressalta-se que os quadrados com contorno amarelo são as entradas da rede, enquanto os quadrados com contorno vermelho são os gabaritos para a saída da rede.



**Figure 2. Região analisada no dia 2/1/2961**

Abordando mais detalhes sobre o pré-processamento dos dados, é importante dizer que todos os dados de treinamento e de teste foram normalizados a fim de diminuir a grande disparidade entre os valores, como também para aumentar a qualidade das previsões da rede. Outrossim, a separação dos dados de treino e teste foi realizada da seguinte maneira: os dados que datam entre 1961 e 1980 foram utilizados para o treinamento da rede, enquanto que os dados de 2000 até 2022 serviram para o teste da rede neural. Dessa forma, foram utilizados um total de um milhão e quinhentos dados de treinamento.

### 3. Apresentando o Problema

Imagine a existência de 9 pontos sobre nossa malha pré-definida na figura (1), os quais devem formar o quadrado  $3 \times 3$  exemplificado na figura (2). Como é possível observar tem-se 4 pontos de contorno amarelo e 5 pontos de contorno vermelho, cada um deles com uma latitude e uma longitude determinadas. Além disso, cabe acrescentar que, em determinada data, os valores da quantidade de chuva, em milímetros, são conhecidos nos pontos amarelos. Sabendo dessa informação, e considerando a mesma data anterior, quer-se determinar a quantidade de chuva, também em milímetros, nos pontos vermelhos.

Portanto, das informações acima, deve-se construir uma rede neural de regressão, a qual recebe como entrada um vetor com 4 números quaterniônicos; visto que, cada um dos 4 pontos pretos armazenam a latitude, longitude, data (mês) e a precipitação (em milímetros). Outrossim, como se quer prever os valores das precipitações nos 5 pontos, já pré-estabelecidos, nossa rede neural deve devolver como saída 5 valores reais.

### 4. Arquitetura da Rede Neural

Como é possível observar no código abaixo, neste projeto, foi considerada uma rede neural densa com, respectivamente, 20, 300, 400 e 366 neurônios

quaterniônicos nas primeiras 4 camadas. Além disso, a camada de saída possui 5 neurônios reais.

```
1 Mlp_Q = Sequential([
2     kr.Input(shape=(16,)),
3     V_Dense(algebra=quaternions, units=20, activation='relu'),
4     V_Dense(algebra=quaternions, units=300, activation='relu'),
5     V_Dense(algebra=quaternions, units=400, activation='relu'),
6     V_Dense(algebra=quaternions, units=366, activation='relu'),
7     Dense(units=5, activation=None, use_bias=False)])
```

**Listing 1. Configurações de Treinamento**

Cabe ressaltar que o formato de entrada da rede corresponde a um vetor com 4 números quaterniônicos. Porém, como é possível ver no código acima, foi realizado um flatten nos vetores de entrada, o que resultou em entradas no formato (16,). Outrossim, como consequência do formato da camada de saída, obtêm-se um vetor com 5 números reais na saída da rede.

## 5. Aspectos sobre o Treinamento

O grupo treinou a rede por 400 épocas com as seguintes configurações do código abaixo. Note que, como tem-se uma rede de regressão, foram utilizados o "Erro Quadrático Médio (mse)" e a "Raiz do Erro Quadrático Médio" como, respectivamente, a loss function e a métrica da rede.

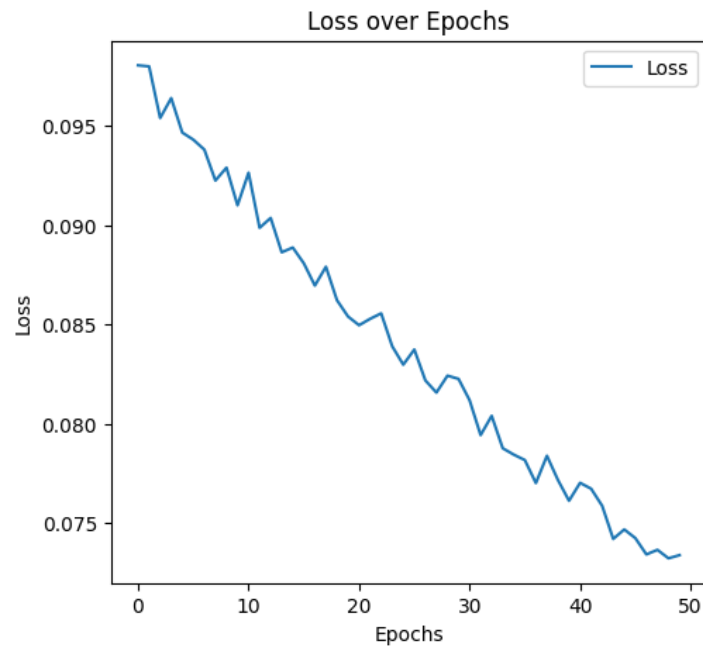
```
1 Mlp_Q.compile(loss='mse', optimizer='adam', metrics=[kr.metrics.
    RootMeanSquaredError()])
```

**Listing 2. Arquitetura da Rede Neural**

Uma informação importante a ser considerada é que, como o treinamento foi realizado utilizando a GPU do Colab, para que não houvesse o risco de o Google derrubar a sessão, o treino foi sendo realizado de 50 em 50 épocas no início e, posteriormente, de 100 em 100 épocas, sendo que os arquivos de treino foram sendo salvos à medida que os treinamentos parciais iam terminando.

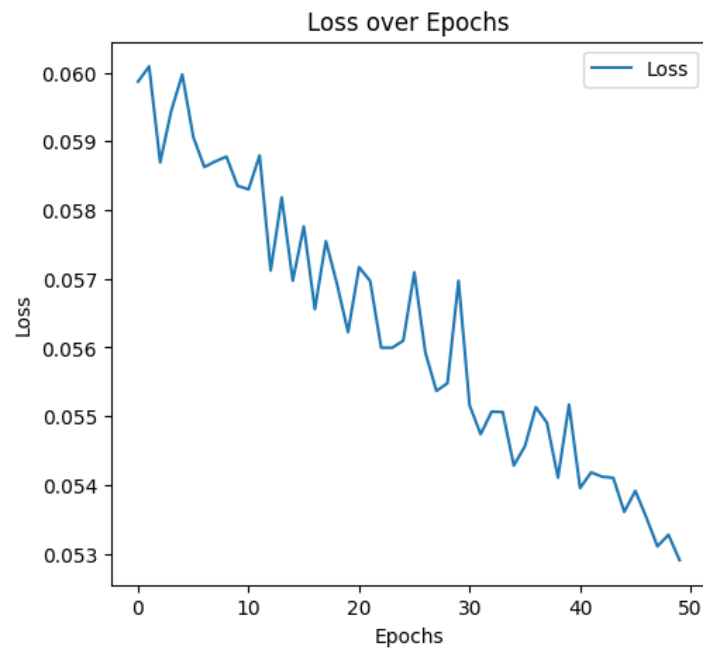
Ao longo do treinamento também foram salvos alguns arquivos de *history*, que armazenam o comportamento da loss sobre os dados de treinamento à medida em que se aumenta o número de épocas. Veja abaixo alguns gráficos que mostram o comportamento da loss:

- **Comportamento da loss entre 50 e 100 épocas:**



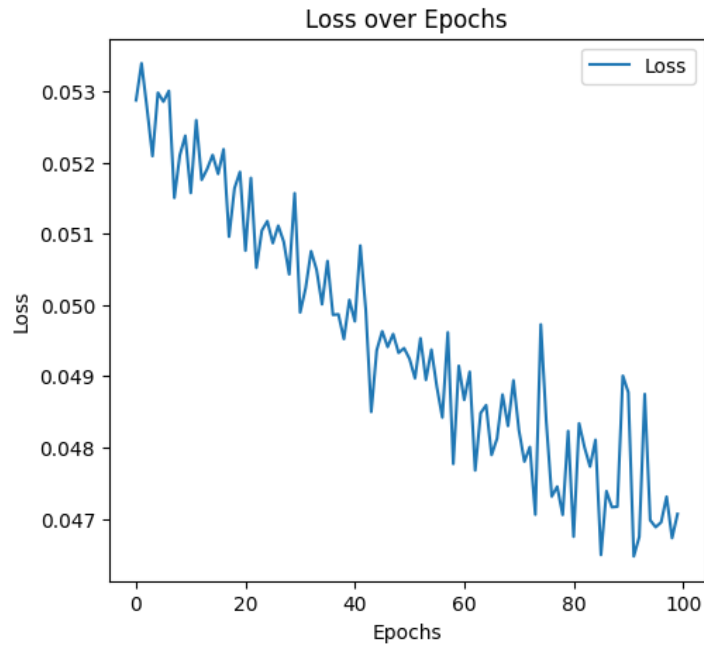
**Figure 3. Loss do treinamento realizado da época 50 até a época 100**

- **Comportamento da loss entre 150 e 200 épocas:**



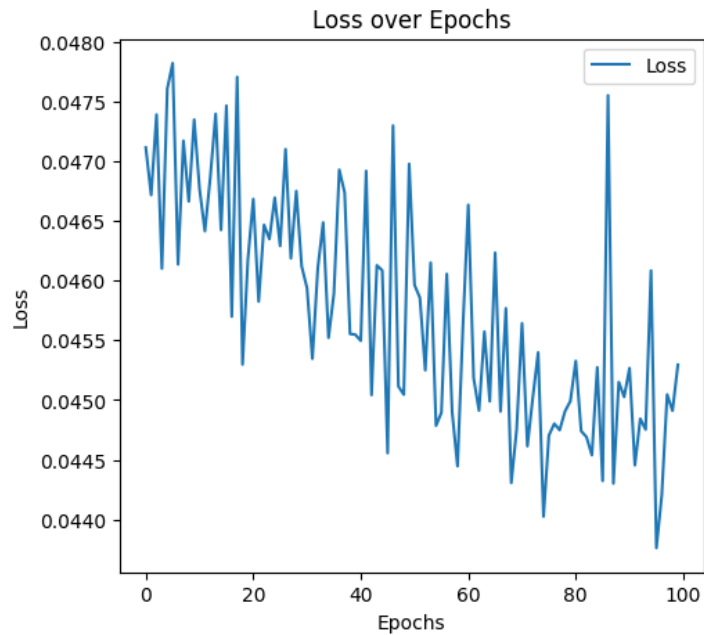
**Figure 4. Loss do treinamento realizado da época 150 até a época 200**

- **Comportamento da loss entre 200 e 300 épocas:**



**Figure 5. Loss do treinamento realizado da época 200 até a época 300**

- **Comportamento da loss entre 300 e 400 épocas:**



**Figure 6. Loss do treinamento realizado da época 200 até a época 300**

Como é possível verificar nos gráficos acima, inicialmente, a loss vai caindo bem, entretanto, à medida que aumentamos o número de épocas, ocorre o aparecimento de uma instabilidade no processo do decaimento da loss, o que dificulta o treinamento. Por esse motivo, como a instabilidade sempre cresce com o

aumento do número de épocas, ficou decidido que o treinamento fosse encerrado em 400, visto que a instabilidade já estava bem grande nesse ponto do treino.

## 6. Resultados

Para avaliar o desempenho da rede neural no conjunto de teste, a rede foi rodada, para diversas datas distintas, considerando a grade, de latitudes e longitudes, pré-definida na figura (1). Após essa etapa, foi possível medir, para cada uma das datas, o erro cometido pela rede neural em cada ponto da malha, isto é, foram calculados os erros  $E = \hat{V} - V$  para cada coordenada, onde  $\hat{V}$  é a quantidade de chuva verdadeira e  $V$  é a estimativa da rede.

Como foi dito, a rede neural foi rodada diversas vezes na mesma malha para diversas datas distintas. Por esse motivo, uma boa forma de verificar o desempenho da rede é plotar o gráfico do erro médio cometido pela rede em cada coordenada, isto é, se a rede foi executada para  $k$  datas distintas, então devemos plotar o gráfico de  $E_{médio} = \frac{1}{k} \sum_{i=1}^k (\hat{V}_k - V_k)$  para cada coordenada. Veja o gráfico do  $E_{médio}$  em cada coordenada da malha na figura (7).

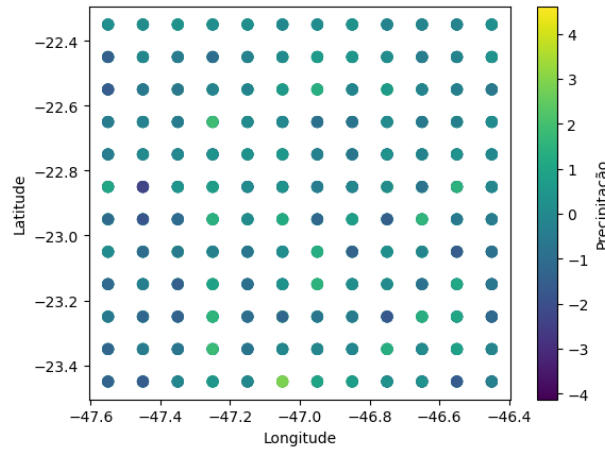
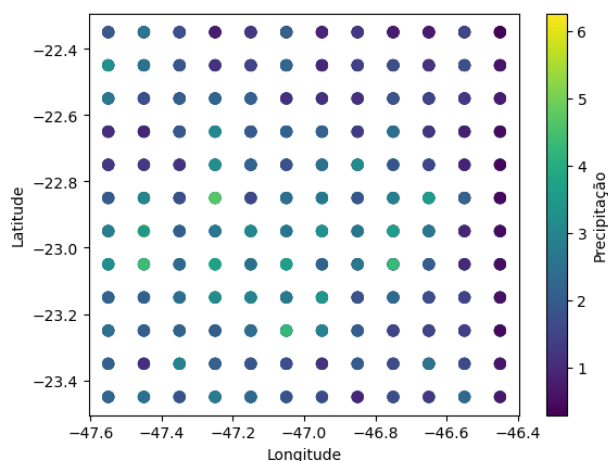


Figure 7. Gráfico do Erro médio em cada coordenada

Como é possível observar no gráfico da figura (7), os erros médios ficaram bem baixos, com a maioria dos pontos em torno de  $0mm$  e com apenas alguns pontos com erro médio de  $4mm$  ou  $-4mm$ . Porém, no gráfico acima o sinal de cada erro, positivo ou negativo, foi considerado durante a média. Esse fato, certamente, contribuiu para deixar os erros médios mais baixos. Por esse motivo, é interessante plotar o gráfico da média dos módulos dos erros para cada coordenada, ou seja, o gráfico de  $E_{Módulos} = \frac{1}{k} \sum_{i=1}^k |\hat{V}_k - V_k|$  para cada ponto da malha. Veja o resultado na figura (8).

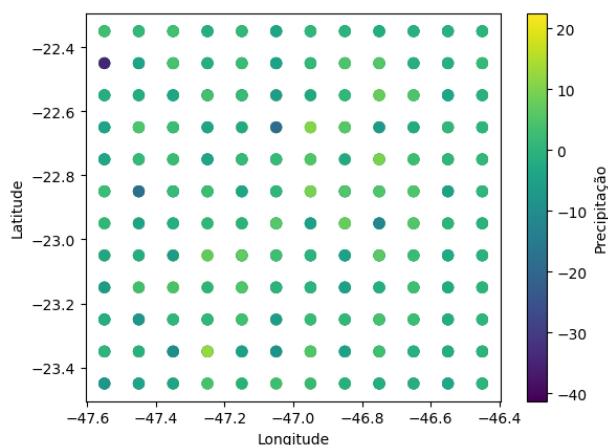


**Figure 8. Gráfico da média dos módulos dos erros em cada coordenada**

Como é possível verificar na figura (8), os resultados também ficaram muito bons. Em geral, exceto por algumas coordenadas, os erros ficaram em torno de 1, 2 ou 3mm. Por conta disso, pode-se concluir que os resultados obtidos pela rede neural proposta neste trabalho ficaram melhores do que havia sido imaginado, visto que, pelo fato de este projeto lidar com fenômenos atmosféricos, o grupo suspeitava que poderiam haver problemas com relação ao caos.

Uma justificativa para o caos não ter interferido, ou não ter interferido o tanto quanto foi imaginado, está no fato de que todas as previsões realizadas em cada um dos 5 pontos vermelhos da figura (2) foram realizadas na mesma data que os 4 pontos amarelos, também da imagem (2).

Ainda é importante considerar, para a medida da qualidade das previsões da rede, como ficou o gráfico do  $E_{médio}$  para a data com pior desempenho possível. Essa visualização é importante para se saber o quanto a rede pode estar errada em suas previsões. Veja as imagens abaixo:



**Figure 9. Gráfico do Erro em cada coordenada para a pior malha obtida**



Ao se observar os gráficos da figura (9), nota-se que a maior parte dos erros ficou entre -5mm e 5mm, visto que existem poucos pontos que destoam muito desse intervalo. Por esses motivos, pode-se concluir que, considerando as latitudes e longitudes em geral, a rede também não teve um desempenho tão ruim nessa pior malha.

## 7. Conclusão

Como foi possível verificar ao longo do projeto, em geral, os resultados obtidos foram bons e muito melhores do que era esperado pelo grupo, considerando que a malha usada foi pequena e o poder computacional foi limitado ao colab. Por conta desse fato, fica claro que a utilização de redes neurais para a realização do refinamento de malhas já existentes é uma boa alternativa ao tradicional método de interpolação, o qual é largamente utilizado nos dias de hoje.

Outrossim, cabe ressaltar que há muito espaço para o aprimoramento da rede neural utilizada neste projeto. Uma boa forma de dar continuidade a este trabalho seria a realização de testes em uma rede neural convolucional, testar diferentes álgebras hipercomplexas para considerar mais informações geográficas, aumentar os pontos de treino para ver como a rede se comporta com um grande espaço amostral e com outros dados climáticos como temperatura e umidade.

## References

- Alexandre C. Xavier, Bridget R. Scanlon, C. W. K. A. I. A. (2022). New improved brazilian daily weather gridded data (1961–2020). *International Journal of climatology*.
- Eyring, V., Bony, S., Meehl, G. A., Senior, C. A., Stevens, B., Stouffer, R. J., and Taylor, K. E. (2016). Overview of the coupled model intercomparison project phase 6 (cmip6) experimental design and organization. *Geoscientific Model Development*, 9(5):1937–1958.