Estimativas da incerteza para propriedades de materiais usando a CGCNN

Técnicas Avançadas de Análise de Dados

Alexandre Duarte
João Fernandes

Universidade de Coimbra

Janeiro, 2023

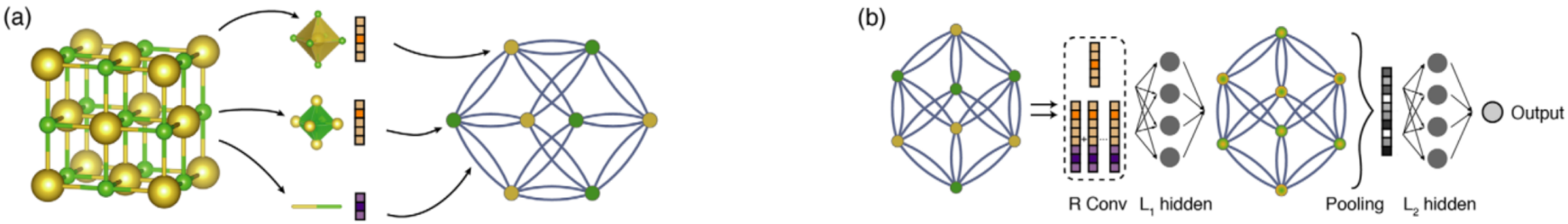
Objetivo

Métodos de Machine Learning estão a ser usados no projeto de novos materiais devido à sua capacidade de prever um valor perto das propriedades do material obtidas por cálculos tradicionais. Isto diminui o numero de cálculos tradicionais (de natureza computacional mais intensa), o que aumenta significativamente a velocidade da análise de possíveis materiais sem a necessidade da sua síntese.

O objetivo deste trabalho foi avaliar se a inclusão da estimativa dessa incerteza agrega informações de valor aos dados. Para esse fim, treinámos dois modelos para prever o módulo de cisalhamento de vários materiais, com um incluindo a previsão de incerteza.

Introdução

O algoritmo base usado foi o Crystal Graph Convolutional Neural Networks (CGCNN).



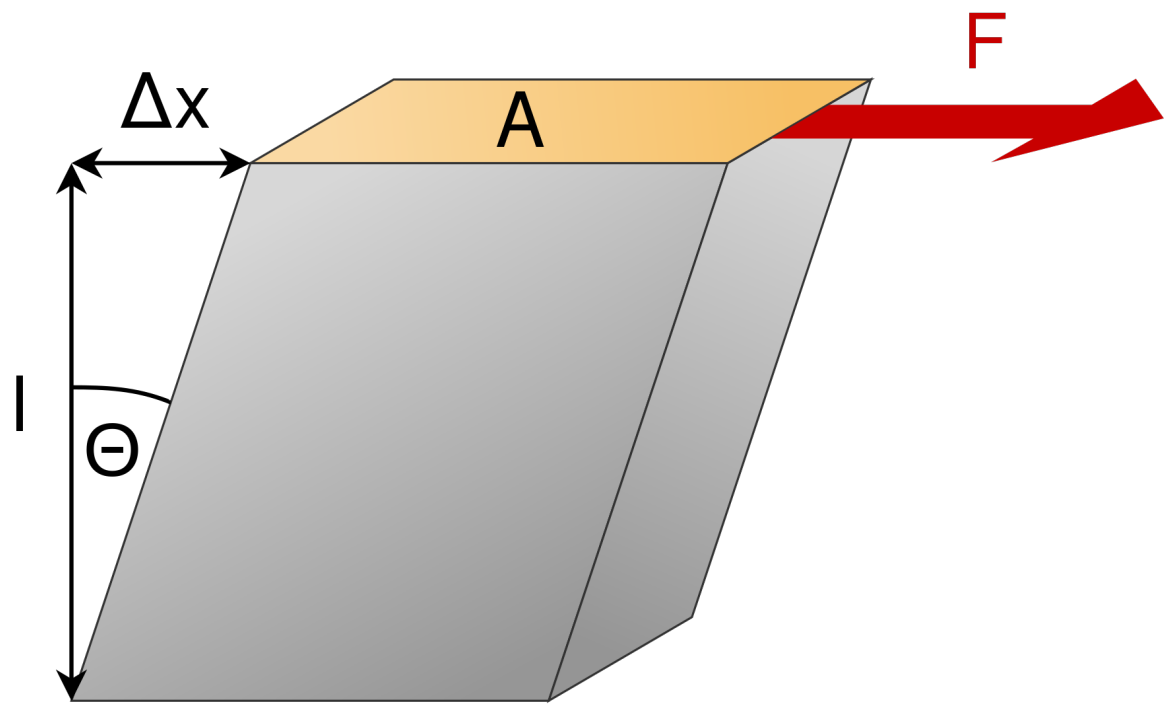
A estrutura cristalina é representada como um gráfico, onde cada átomo é um nó e as arestas representam as ligações químicas entre os átomos. O gráfico de cristal também codifica informações sobre as simetrias do cristal, como o grupo espacial e os vetores de rede.

Dataset

- Data Source
  - The Materials Project database
  - The Perovskite database
- Data Description
  O conjunto de dados inclui informações sobre as propriedades físicas de diferentes materiais, incluindo módulo de cisalhamento, módulo volumétrico e dureza Vickers. Os dados incluem os seguintes campos:
  - 'structure': Informação sobre a estrutura cristalina do material.
  - 'G': O módulo de cisalhamento do material.
  - 'K': O módulo volumétrico.
  - 'Hv': Dureza de Vickers.
  - 'formula': A fórmula química da substância.
  - 'spg': O grupo cristalográfico.
- Data Size
  O dataset incluí informação sobre cerca de 10^4 materiais.

Módulo de cisalhamento

O módulo de cisalhamento, G, está relacionado com a resposta de um corpo a stress torcional, em termos de tensão.
Envolve mudança na forma do corpo sem mudança no seu volume.

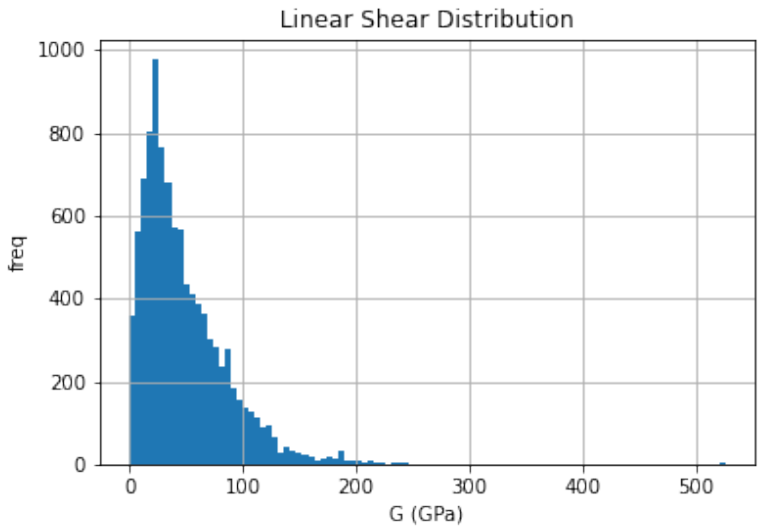


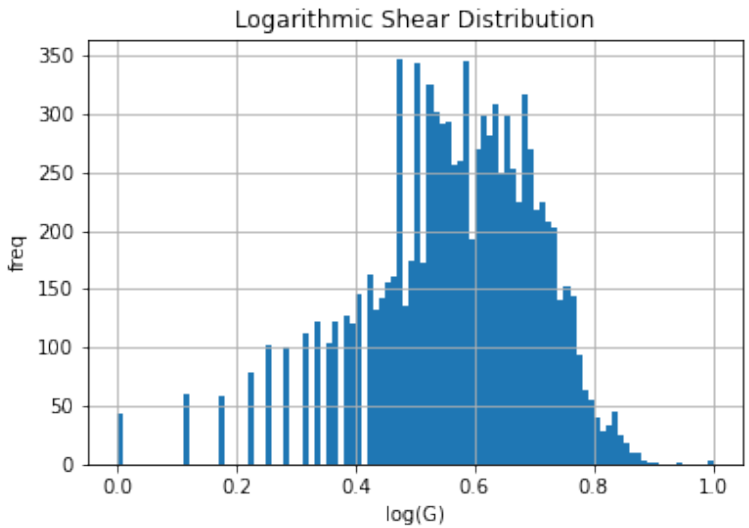
Dataset Analysis

A tabela contém informação estatística sobre o dataset.

	G	K	Hv
count	10000.00000	10000.000000	10000.000000
mean	48.67790	100.534200	6.543098
std	39.19774	69.117557	5.432998
min	0.00000	1.000000	0.024603
25%	21.00000	47.000000	2.885959
50%	38.00000	85.000000	5.175665
75%	67.00000	144.000000	8.904438
max	525.00000	575.000000	99.706853

Shear distributions





Conclusões que se podem tirar sobre a forma com os dados estão distribuídos:
Balanceamento do dataset: A média de G é significamente maior do que a mediana, sendo ambas valores pequenos no intervalo de valores, o que signica que existem materiais com valores elevados que se afastam da população principal, como se pode observar nos histogramas.
Estratégia utilizada para a normalização/escalamento: A transformação da escala linear para logarítmica permite obter uma distribuição mais uniforme ao longo da gama de valores de G. Os dados são distribuídos de maneira mais uniforme e, por isso, é menos provável que o modelo seja influenciado por valores discrepantes ou pontos de dados anómalos.
Problemas encontrados na análise do dataset: Os valores nulos de G foram descartados na conversão da escala linear para logarítmica.

Correr os modelos

Executámos os modelos CGCNN com o conjunto de dados pré-processado para prever o cisalhamento dos materiais, e fizemos isso para 72 combinações diferentes de valores de hiperparâmetros, cada uma com um train size de 80%, e tamanhos do test e evaluation de 10%. Correu-se cada combinação com 200 épocas. Os hiperparâmetros testados foram

- número de convolution layers, responsáveis por aprender e extrair características dos dados.
- número de hidden layers, responsáveis por pegar nos recursos extraídos das convolution layers e usá-los para fazer previsões sobre os dados.
- comprimento das hidden layers, que se refere ao número de features da hidden layer.

Funções de Loss

Durante o treino, o modelo ajusta os seus parâmetros internos com o objetivo de minimizar a função de loss.
Para a tarefa de regressão no primeiro modelo, a função de loss usada foi a Mean Squared Error loss, enquanto que para a tarefa de distribuição no segundo modelo, a função de loss usada foi a Gaussian Negative Log Likelihood loss.

Mean Squared Error

A Mean Squared Error loss é dada por

$$loss = \frac{1}{N}\sum_{i=1}^{n}(x_n - y_n)^2 = 1 \tag{1}$$

em que x_n e y_n são a predição do modelo e o valor alvo, respetivamente, com N o número total de elementos.

- Vantagens
  - fácil de calcular e interpretar
- Desvatagens
  - Sensível a outliers, um único grande erro pode ter um grande impacto
  - Assume que os erros têm uma distribuição normal

Loss function- Gaussian Negative Log Likelihood

A Gaussian Negative Log Likelihood loss é dada por

$$loss = \frac{1}{2}(log(max(var, eps)) + \frac{(x_n - y_n)^2}{max(var, eps)}) + const \qquad (2)$$

em que var é a variância e eps é um parâmetro utilizado para a estabilidade numérica da função.

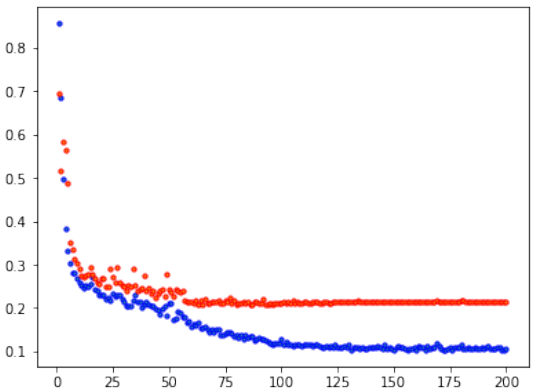
- Vantages
  - Menos sensível a outliers, pois coloca menos peso em erros individuais
  - Não assume que os erros têm uma distribuição normal
- Disadvantages
  - Maior carga computacional

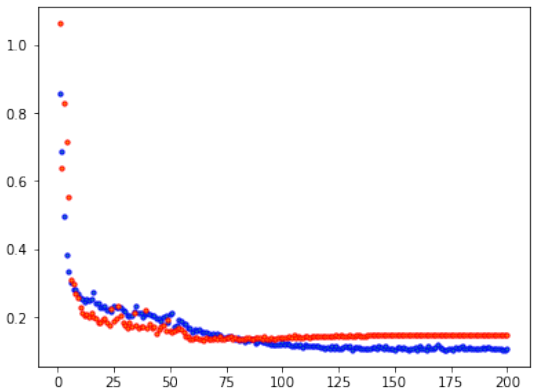
Após correr os modelos, converteram-se os resultados de valores logarítmicos para valores em GPa. Aplicou-se a seguinte fórmula de propagação de erros para a conversão dos desvios padrão, no modelo da distribuição:

$$\delta\sigma = e^{G_{Log}}\sigma_{G_{Log}} \tag{3}$$

Loss function minimum- Regression model

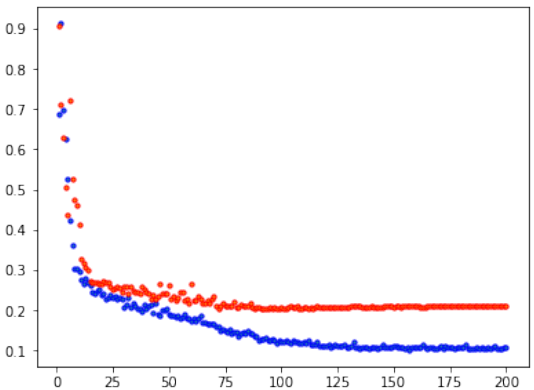
4 hidden layers, 128 hidden features, 2 convolutional layers

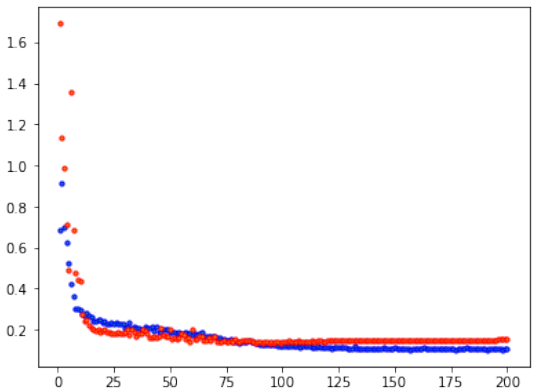




Mean Absolute Error minimum - Regression model

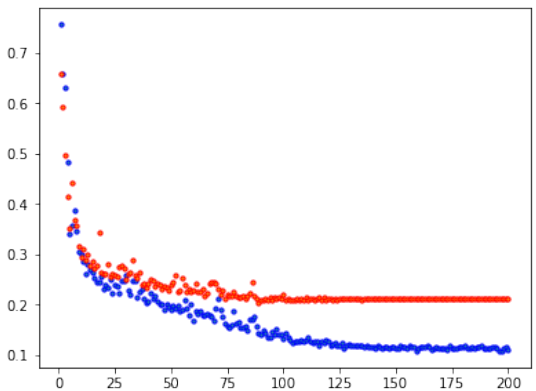
3 hidden layers, 128 hidden features, 2 convolutional layers

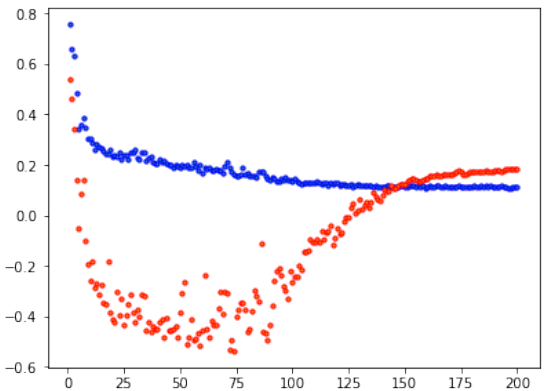




Loss function minimum- Distribution model

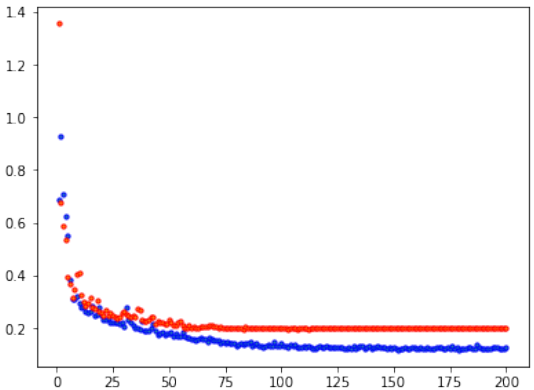
5 hidden layers, 16 hidden features, 4 convolutional layers

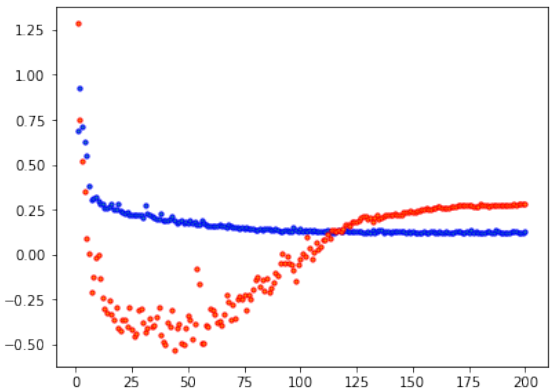




Mean Absolute Error minimum - Distribution model

1 hidden layers, 128 hidden features, 4 convolutional layers





Regression model

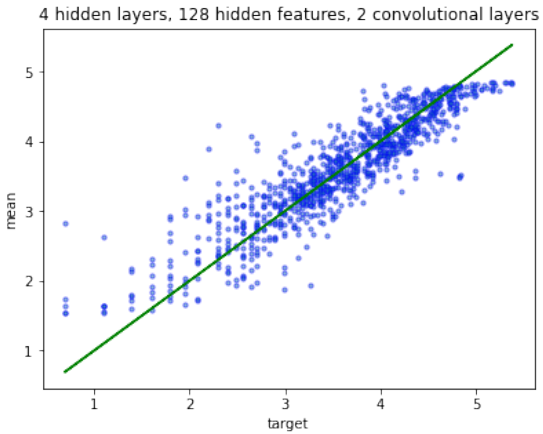
minLoss	minMAE
87 epochs	64 epochs
0.240	0.226

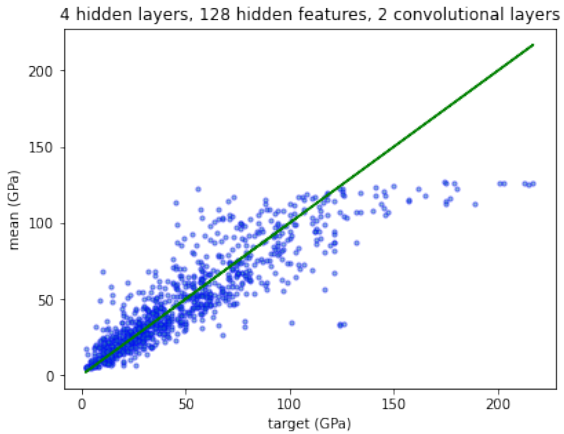
Distribution model

minLoss	minMAE
74 epochs	82 epochs
0.240	0.228

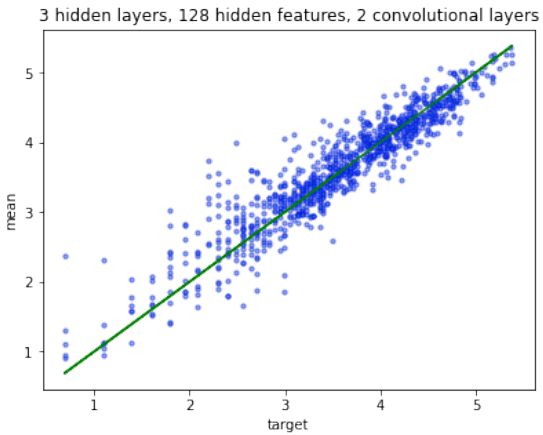
Melhores resultados obtidos para o modelo de regressão

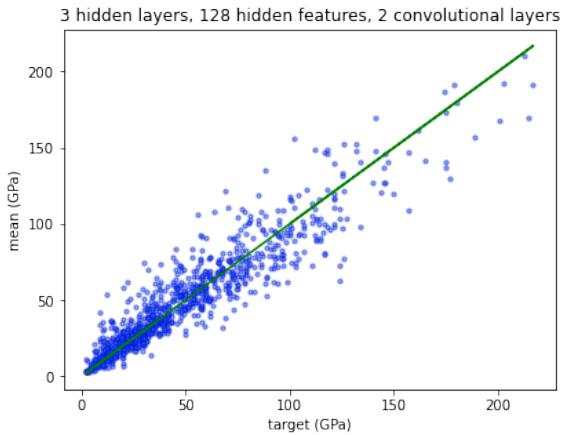
Mínima Loss:





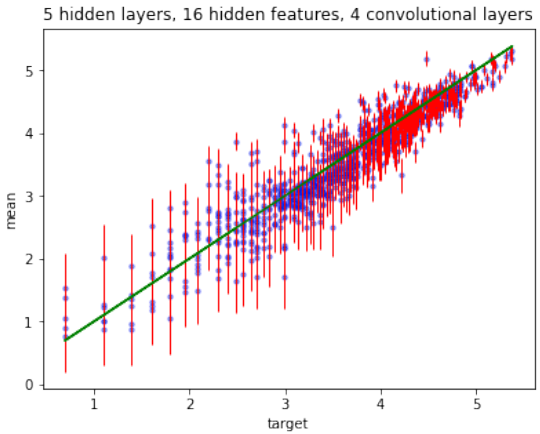
Mínimo MAE:

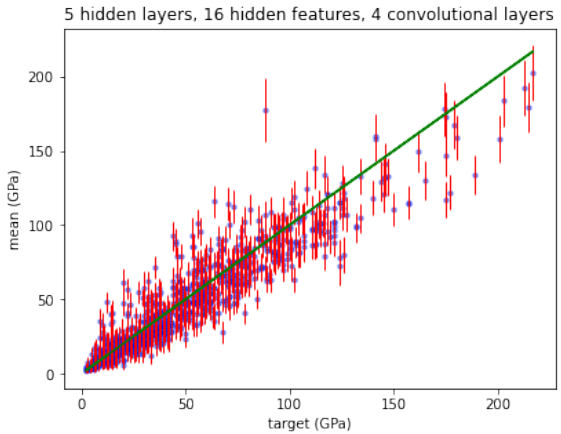




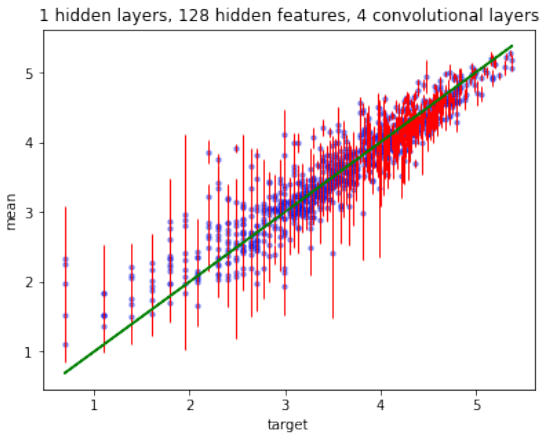
Melhores resultados obtidos para o modelo de distribuição

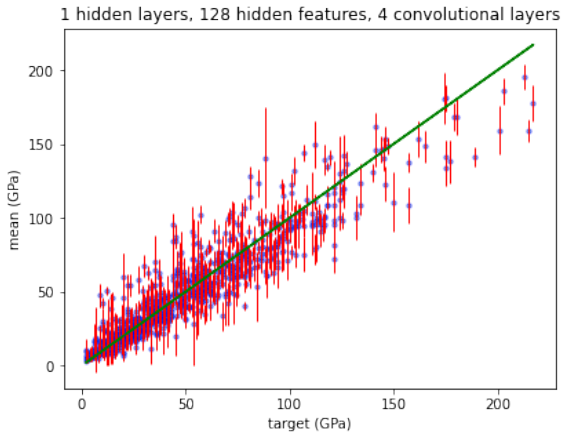
Mínima Loss:





Mínima MAE:





Regression model

minLoss	minMAE
5.326%	5.734%

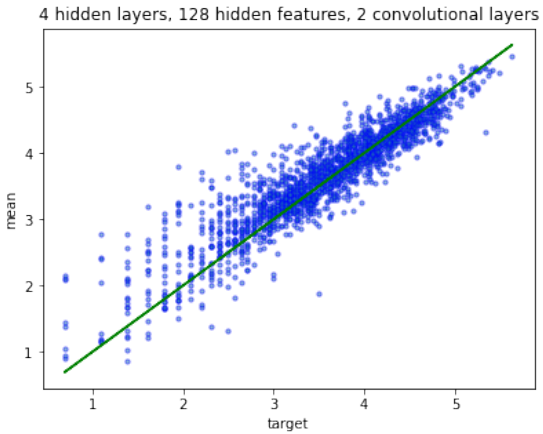
Distribution model

minLoss	minMAE
50.70%	41.55%

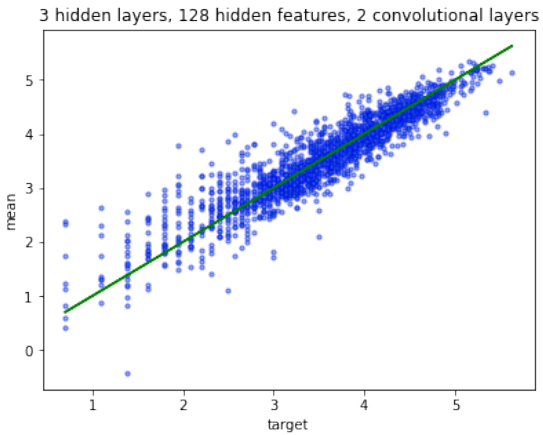
Obteram-se as épocas em que a função loss é mínima para todos os conjuntos de hiperparâmetros acima e correram-se mais uma vez os modelos com os números de épocas obtidas para um train size de 60%, test size e evaluation size de 20%.

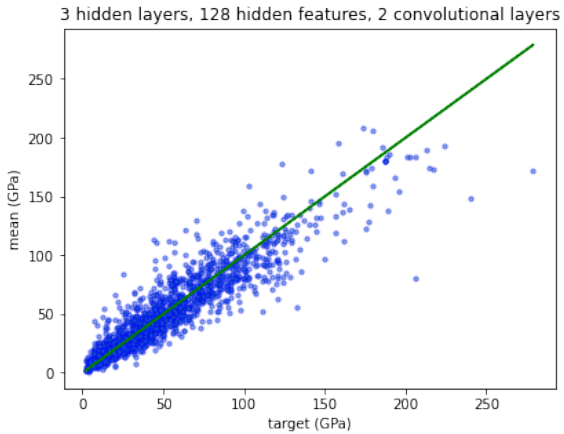
Melhores resultados obtidos para o modelo de regressão

Mínima Loss:



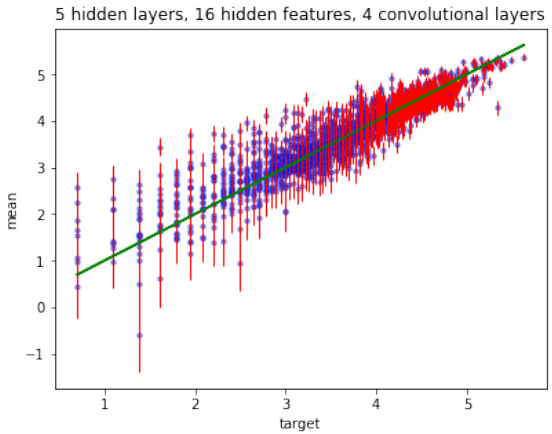
Mínimo MAE:

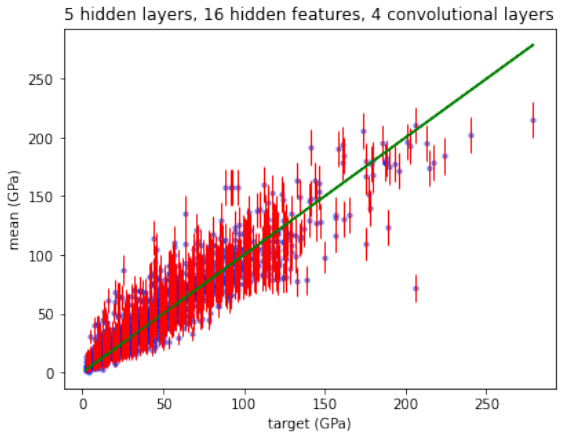




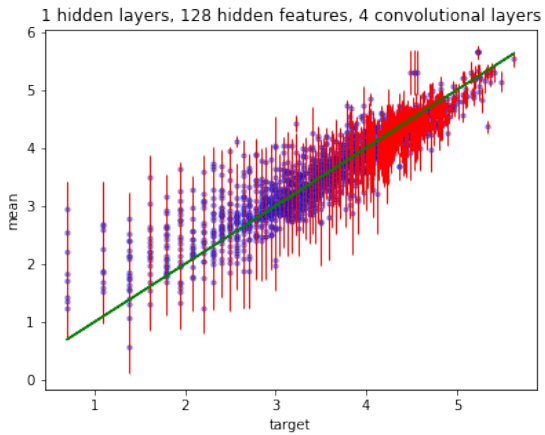
Melhores resultados obtidos para o modelo de distribuição

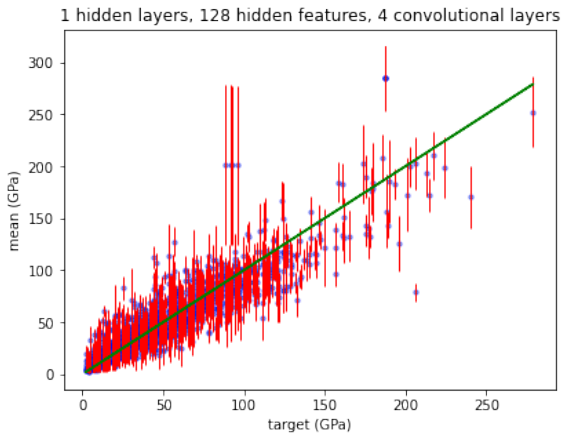
Mínima Loss:





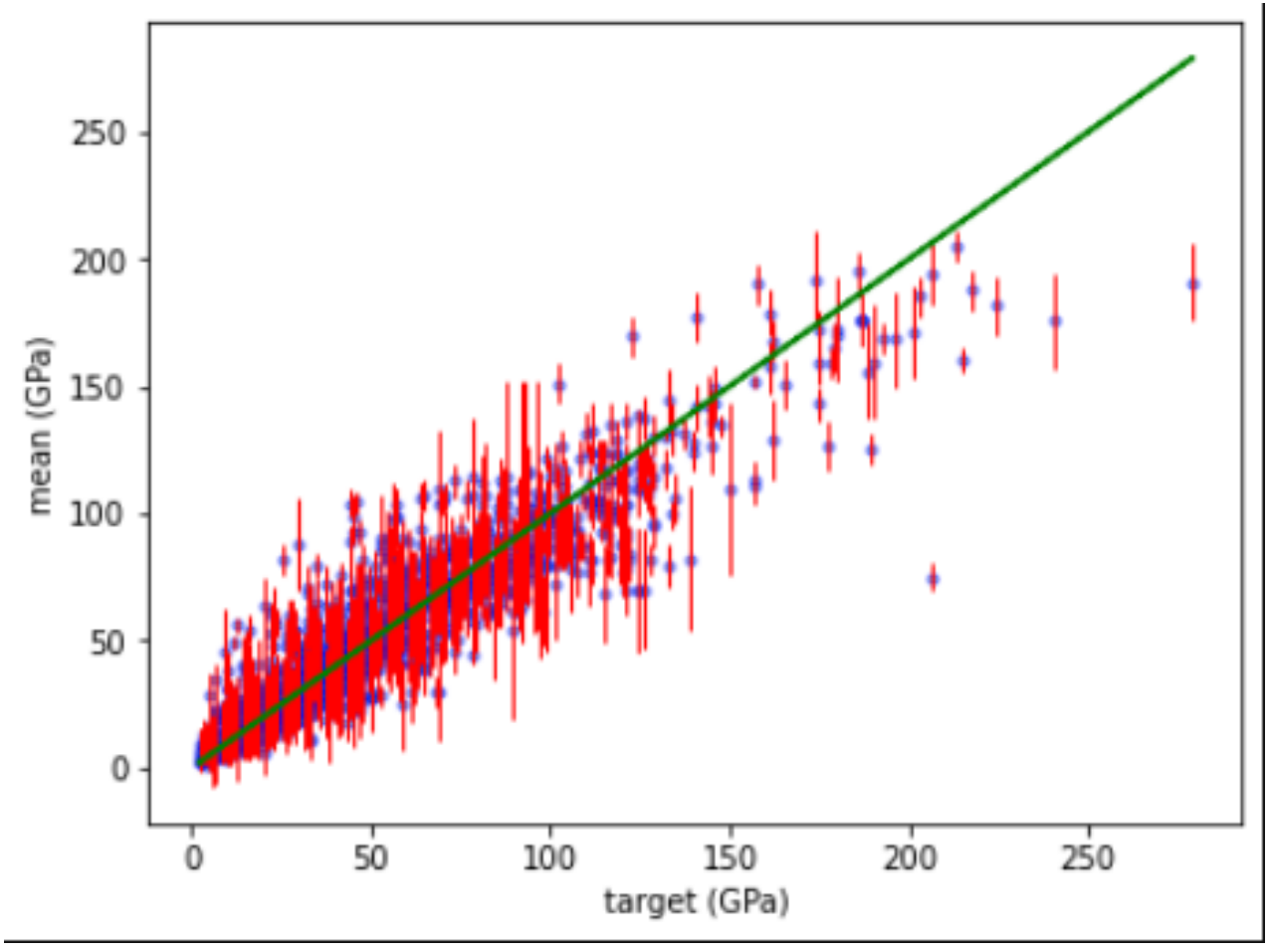
Mínima MAE:





Conclusão

Utilizando o MAE como desvio padrão do modelo de regressão, é possível verificar que a fração de targets que está dentro do intervalo de confiança é muito menor no modelo de regressão do que no modelo de distribuição, o que permite concluir que o segundo modelo é mais eficiente do que o primeiro modelo na deteminação do módulo de cisalhamento dos materiais.


mean (GPa)
target (GPa)
0
50
100
150
200
250