Relatório 1: Linear vs MLP

João Victor de Souza Albuquerque

[1]Instituto Federal de Educação, Ciência e Tecnologia do Ceará - Campus Maracanaú (IFCE)
Av.Parque Central, 1315 - Distrito Industrial I, Maracanaú - CE, 61939-140

Resumo. *Este relatório documenta o experimento do primeiro trabalho de rna.*

1. Base Teórica

1.1. Regressão Polinomial

Para muitos problemas de regressão uma função linear não é capaz de satisfatoriamente se adequar aos dados, assim não sendo capaz de produzir predições adequadas para o problema. Tendo em vista essa questão, uma das soluções encontradas foi aplicar transformações não lineares a aos dados conhecidos, com isso criando novas colunas com dados não-lineares. Na regressão polinomial a transformação não-linear é utilizada para aumentar o polinômio da equação de predição, com isso permitindo criar funções que se adéquem melhor aos dados de treino. Dessa forma, para um polinômio de ordem N:

$$\hat{y}_i = w^T x_i, \tag{1}$$

$$x_i = \begin{bmatrix} 1 & x_i & x_i^2 & \cdots & x_i^P \end{bmatrix}^T, \tag{2}$$

$$w = \begin{bmatrix} w_0 & w_1 & w_2 & \cdots & w_P \end{bmatrix}^T. \tag{3}$$

Com a aplicação da não-linearidade somos capaz de obter uma reta de regressão mais flexível que aumenta o poder de generalização do modelo, que é a capacidade do modelo a se adaptar a dados não vistos, porem essa flexilidade também possibilita a ocorrência de dois outros fenômenos que são negativos para a generalização.

O primeiro desses fenômenos é o underfitting (subajuste) que ocorre quando o modelo não tem expressividade para se ajustar aos dados de treino. O outro fenômeno é o overfitting que ocorre quando o modelo se ajusta de mais ao dados de treino.

1.2. Regularização

A regularização foi uma ideia que surgiu tendo em vista o problema de overfitting, buscando a capacidade de modelos serem mais flexíveis mantendo a generalização. Para isso ocorrer é preciso adicionar na função de custo, um termo proporcional à norma quadrática dos parâmetros.

$$\mathcal{J}(w) = \frac{1}{2}(y - Xw)^T(y - Xw) + \frac{\lambda}{2}||w||^2$$

$$||w||^2 = w^T w = \sum_{d=1}^{D} w_d^2$$

O método de otimização utilizado modelo polinomial desse trabalho foi o OLS e a regularização é aplicada nesse método da seguinte forma:

$$w = (X^T X + \lambda I)^{-1} X^T y.$$

1.3. Normalização

As vezes os dados que queremos que o modelo se adapte estão em escalas muito diferentes o que pode acarretar em dificuldade do modelo em se adaptar a eles, por isso é recomendado deixar os dados numa escala comum, esse processo se chama de normalização, que pode ser colocar os dados em uma escala de [0,1] ou [-1,1] ou mesmo forção uma média e variância. Esse método de preprocessamento nos da maior controle dos valores dos parâmetros e facilita o ajuste dos hiperparâmetros

No trabalho foi utilizado a normalização dos dados na escala de [0,1], essa regularização seguir os seguintes passos:

Primeiro se calcula o valor máximo do vetor Y e matriz X, coluna a coluna:

$$y_{max} = \max(y), \quad [x_{max}]_d = \max([X]_d), \forall d.$$

Depois disso, calcule o menor valor do vetor Y e da matiz X, coluna a coluna:

$$y_{min} = \min(y), \quad [x_{min}]_d = \min([X]_d), \forall d.$$

E por fim se aplica a formula a seguir para finalizar a regularização:

$$y \leftarrow \frac{y - y_{min}}{y_{max} - y_{min}}, \quad [X]_d \leftarrow \frac{[X]_d - [x_{min}]_d}{[x_{max}]_d - [x_{min}]_d}, \forall d.$$

2. Metodologia

2.1. Dados

Importei os conjutos de dados, boston.csv e gauus.csv, utilizando a função read_csv do pandas e salvei eles como dataframes com nomes dos seus repectivos conjuntos. Para facilitar o trabalho com o conjunto de dados da boston eu renomeie suas colunas e nenhum modificação foi necessária no conjunto de gauus.

2.2. Adaptações ao modelo

Para tornar o meu modelo de regressão linear existente em um modelo de regressão polinomial, foi preciso implementar uma função que adiciona não-linearidade ao dataset. A função de transformação polinomial recebe o grau do polinômio e dataset, e aplica a transformção para todas as colunas e no fim devolve um novo dataset adicionado as colunas não-lineares. Essa função de transformação é feita internamente a função fit() do modelo.

Apos a transformação, os dataset é mandado para a função fit_ols() que é a mesma do ultimo trabalho por exceção da regularização.

Para adicionar a regularização foi necessário apenas modificar a linha de calculo dos pesos e montar antes dessa linha uma matriz identidade. Tanto o grau do polinômio quanto o fator de regularização são recebidos na função fit() do modelo, e o valor do grau do polinômio é salvo numa variável interna do modelo.

A função predict() também foi levemente alterada, sendo necessário adicionar a função de transformação polinomial internamente e transformar a linha de calculo de Y em um loop, não é necessário repassar o grau do polinômio pois ele já foi salvo no fit.

2.3. Normalização

Para Implenetar a classe de normalização, preciso desenvolver 3 funções, a função fit(), normalize e desnormalize. A função fit salva os minemos e máximos do X e Y internamente na classe. A função de normalize pega os valores de máximos e minemos salvos na classe e utiliza na função de normalização já mostrada e que foi traduzida para código, e no final devolve os X e Y normalizados. A função desnormalize pega os valores mínimo e máximo salvos internamente na classe e as utiliza na inversa da função de normalização.

2.4. Questão 1

Primeiro separei os dados em conjunto de treino e teste, utilizando a função train_test_split() do sklearn com test_size de 0.2 e random_state de 42. Após isso, separei o conjuto de treino em train_x e train_y, e o conjunto de teste em test_x e test_y.

Com os conjuntos já em mãos, eu defini a classe de normalização e chamei sua função fit() para ajustar a escala aos dado. Em seguida chamei a função normalize para o conjunto de treino e teste, e salvei as saidas em variaveis.

Com os dados já normalizado, eu fiz um loop para a criação dos 11 modelos polinomiais e calcular seu RMSE tanto para o conjunto de treino quanto o de teste. Primeiro o loop define uma instancia do modelo e o treina com os dados normalizado e o grau do polinômio da roda, em seguida é feita a predição para os dados de treino e a saída do modelo é desnormalizada, com a predição do modelo desnomalizada é calculado o RMSE do treino para esse modelo, que salvo numa lista de RMSE de treino para ser utilizado depois, e em seguida é printado o RMSE do treino daquele polinômio para ter controle do processo. Terminado a obtenção do RMSE do treino, é feito em seguida a obtenção do RMSE do teste, que segue os mesmos passos do treino e tem o seu RMSE salvo numa lista de RMSE de teste. Com os RMSE obtidos é plotado um gráfico de por polinômio.

O processo descrito acima é feito duas vezes, a primeira sem utilizar a regularização e a segunda vez usando a regularização com calor de 0,01

2.5. Questão 2

Para fazer a segunda questão eu segui o mesmo passo a passo da primeira questão mudando o numero e grau dos dos polinômios e os dados de treino e teste. E com os dados obtidos fiz o gráfico de RMSE por polinômio e o gráfico de dispersão dos dados com a reta de regressão.

3. Resultado

3.1. Questão 1

Testando os 11 modelos sem regularização obtive o seguinte gráfico da imagem 1. Pode ser observado pelo gráfico que o modelo começou a ter overfitting no modelo polinomial de grau 4 e esse fenômeno atingiu seu pico no modelo de grau 9, pelo gráfico pode se presumir que o melhor modelo é o de 3 ou 4,porem vendo pela tabela de RMSE dos modelos não regularizados,imagem 2, o melhor modelo ficou sendo o de grau 5. Agora testando os 11 modelos com regularização se obteve o gráfico da imagem 3, pode ser observado pelo gráfico que não ocorreu o fenômeno de overfitting com os modelos regularizados como o esperado, além disso se percebe que conforme o grau do polinômio vai aumentando melhor vai ficando o RMSE, portanto nesse cenário o melhor modelo é o de maior grau.

3.2. Questão 2

Como pode ser observado nos gráficos de dispersão com a curva do polinômio sem regularização, conforme o grau do polinômio foi aumentando melhor ele se encaixa na dispersão, e se comparar com os gráficos das curvas dos modelos regularizados se percebe que os modelos sem regularização se adaptam mais rápido e melhor a dispersão que os modelos regularizados. Pelo gráfico de RMSE da imagem 5 se ver que o modelos regularizados não consegui generalizar para os dados de gauss, enquanto se observar o grafico de RMSE da figura 4 vemos que os modelos não regularizados conseguem generalizar para os dados de gauss, portante decidirei com base neles, pelo gráfico e a tabela da RMSE da imagem 6 pode se concluir que o melhor modelo é o de grau 17.

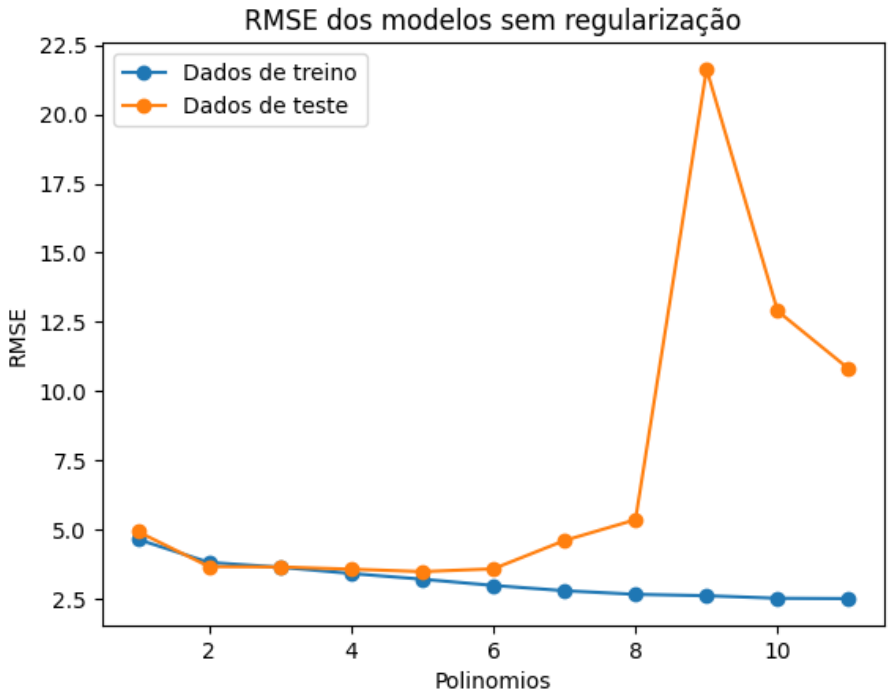

Figure 1. grafico de RMSE

Figure 2. tabela de RMSE

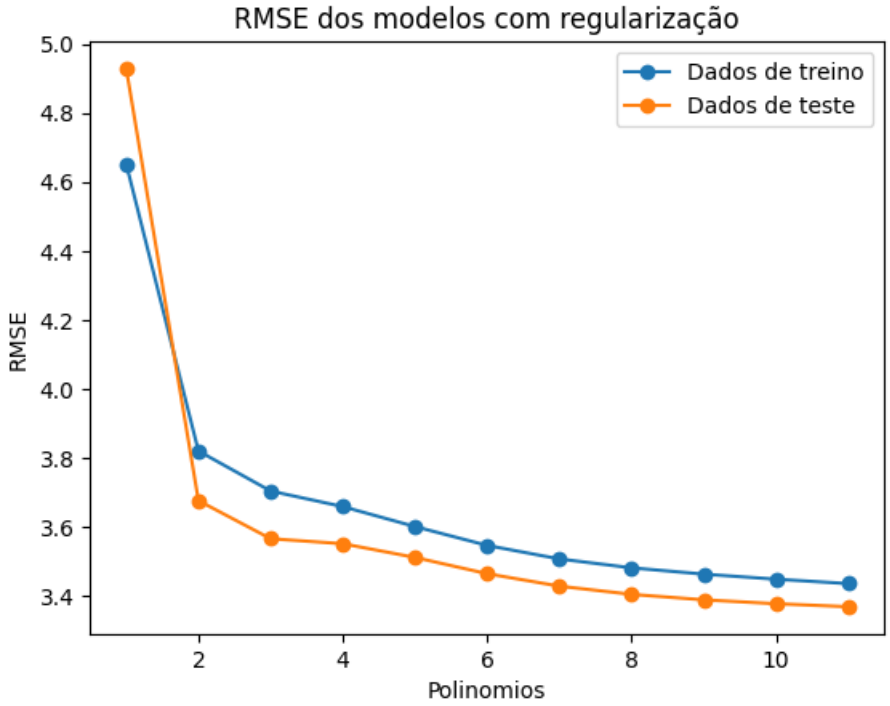

Figure 3. grafico de RMSE

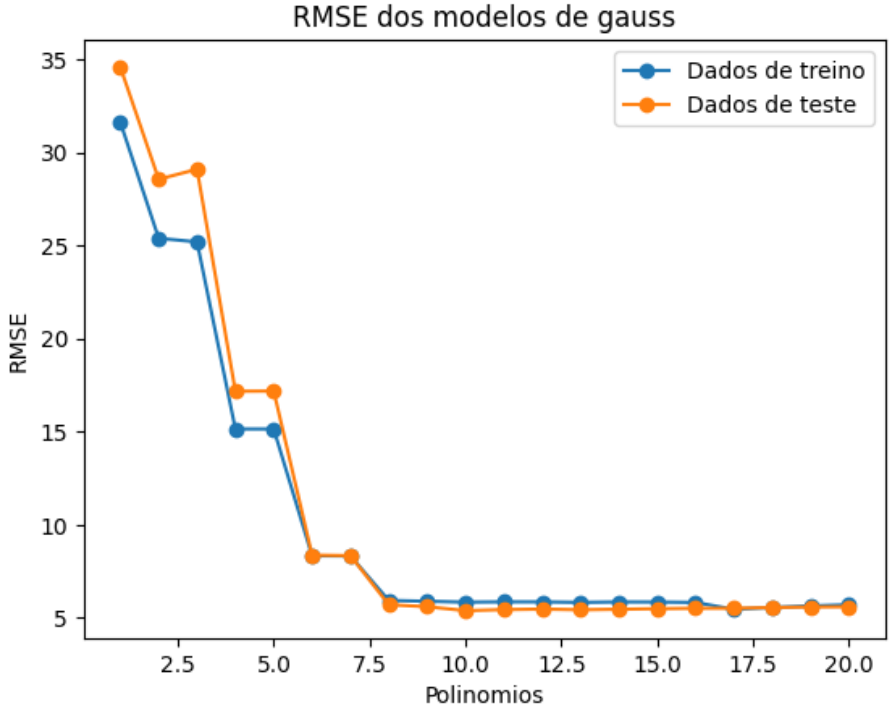

Figure 4. grafico de RMSE

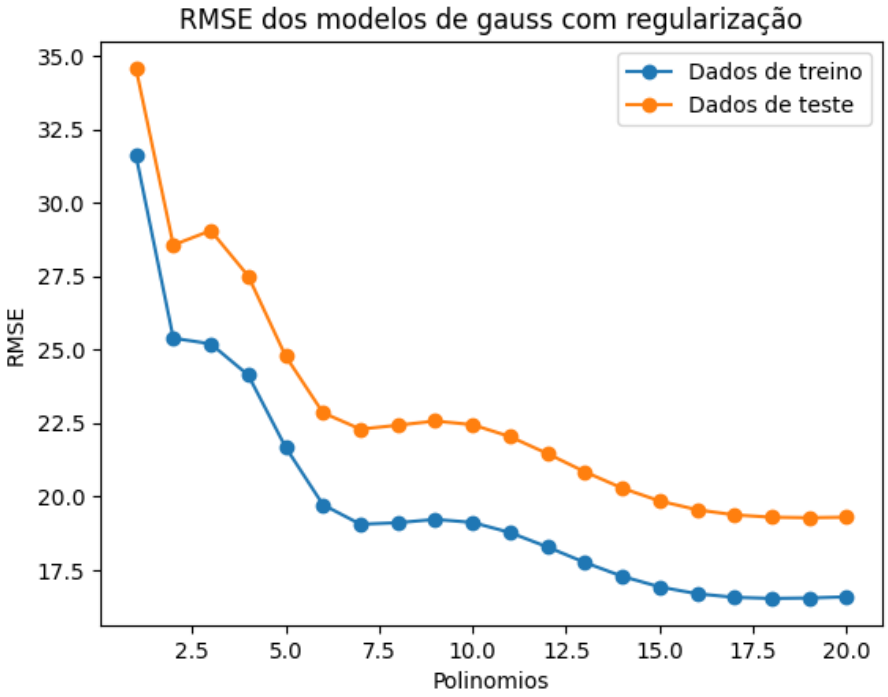

Figure 5. grafico de RMSE

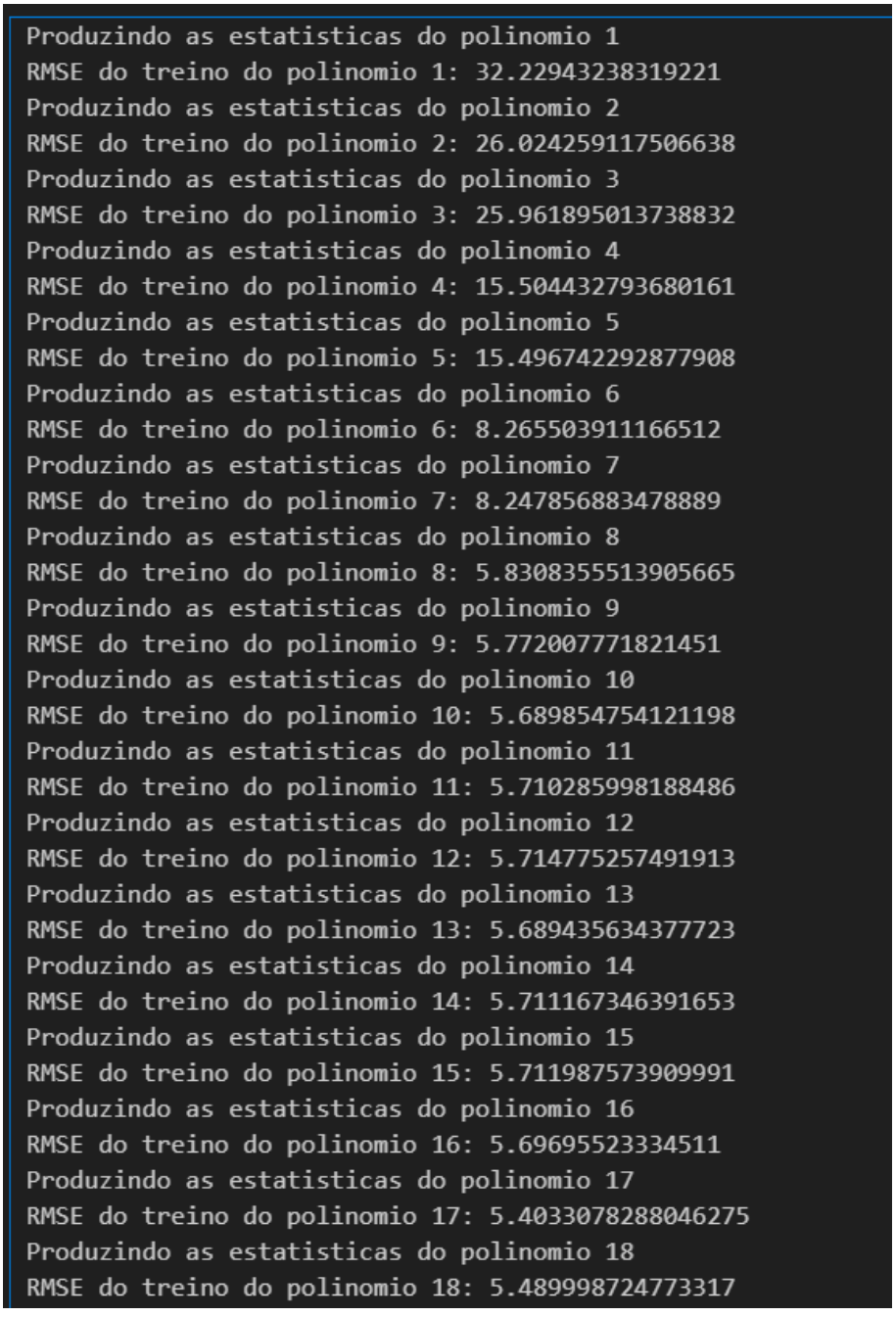

Figure 6. tabela de RMSE