

Redes Neurais Artificiais - Trabalho Intermediário

Pedro Henrique Gomes Mapa da Silva
Aluno de graduação em engenharia elétrica
UFMG
05/07/2022, Belo Horizonte, Brasil

I. INTRODUÇÃO

O treinamento de redes neurais artificiais (RNAs) com projeção fixa na camada intermediária, como Extreme Learning Machines (ELMs) e Radial Basis Functions (RBFs), ou de redes sem projeção fixa como Multilayer perceptron (MLP) é útil para regressões e classificações, mas dependendo de vários fatores, incluindo a representatividade do conjunto de treinamento ao problema, erro de amostragem e a complexidade do modelo, é difícil atingir um modelo que generaliza para o problema. Uma forma de abordar esse problema é a regularização, que leva em conta na função o módulo dos pesos de custo multiplicado por um hiper parâmetro λ escolhido a priori do treinamento e a seleção deste que maximiza a generalização do modelo pode ser difícil. O Objetivo desse trabalho é analisar os resultados da técnica *Leave One Out* (LOO) e da técnica de validação cruzada na seleção do parâmetro λ .

II. REVISÃO DE TRABALHOS CORRELATOS

A. Regularização

Sem qualquer tipo de regularização, temos a função de custo definida como a soma dos erros quadrático

$$J_e = \sum_{i=0}^N (y_i - \hat{y}_i)^2, \quad (1)$$

sendo N o número de amostras, y_i o valor verdadeiro da amostra i e \hat{y}_i o valor previsto para amostra i . No caso de uma camada de projeção fixa \mathbf{H} , a solução analítica da matriz de pesos \mathbf{W} em (1) existe e é dada por

$$\mathbf{W} = \mathbf{H}^{-1} \mathbf{Y}, \quad (2)$$

mas para modelos como MLP, é preciso executar um algoritmo de otimização para encontrar o conjunto de pesos que minimizam o erro. Entretanto, embora os pesos encontrados minimizem o erro no conjunto de amostras em que foram treinados, eles geralmente performam mal para outras amostras que não foram utilizadas no treinamento, causando *overfitting*. Uma abordagem a esse problema é a regularização, que consiste em adicionar o seguinte termo, referente a magnitude dos pesos, à função de custo

$$J_w = \lambda \|\mathbf{W}\|_2, \quad (3)$$

e no erro final temos

$$J = J_e + J_w, \quad (4)$$

que para projeção fixa tem solução analítica para matriz de pesos

$$\mathbf{W} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I}_p)^{-1} \mathbf{H}^T \mathbf{Y}, \quad (5)$$

, mas que para o MLP ainda precisa ser encontrada por otimização. Segundo [6], (4) restringe o espaço de soluções e reduz a magnitude de cada peso individual, o que implica na diminuição da variância do modelo, e, logo, aumenta sua generalização. Mas a relação entre a variância e λ varia para cada conjunto de dados, então, o que é procurado para cada problema é o parâmetro de regularização λ que aumenta a generalidade do modelo, ou seja, que tem um desempenho melhor para novas amostras.

B. Validação cruzada e Erro de Teste

Para descobrir qual será o melhor λ , devemos primeiro sermos capazes de comparar modelos diferentes, e seria interessante obter a performance esperada em novos dados. Para isso, é preciso dividir as amostras em um conjunto de treinamento e um conjunto de teste, onde o primeiro é utilizado para encontrar o λ e o outro é utilizado para estimar a performance esperada do modelo em novas amostras, mas se essa divisão for feita apenas uma vez é possível que a divisão não seja representativa e os dois conjuntos não pertençam a mesma distribuição, resultando em uma má estimativa do erro em novas amostras. Por isso, devemos repetir o processo de divisão diversas vezes, de forma que a probabilidade de ter uma boa estimativa aumentará com o número de repetições.

Ainda assim, não podemos simplesmente treinar os modelos no conjunto de treinamento e comparar os erros de diversos modelos para selecionar o hiper parâmetro, devemos fazer um processo de validação cruzada, como o k-fold ou LOO, e, agora sim, comparar os erros de validação para encontrar o melhor valor de λ .

III. METODOLOGIA

Abaixo está descrito as diferentes técnicas de validação e datasets utilizados em detalhes. Para as redes com camada de projeção fixa (ELM e RBF) o método de validação foi o Leave-one-out (LOO)

A. k-folds

Essa é uma técnica de validação que consiste em dividir o dataset de treinamento em k partes, e utilizar $k-1$ partições para o treinamento e 1 para validação, repetindo de forma que cada partição será utilizada como validação uma vez, e no final fazer a média dos erros de validação, garantindo que cada amostra

terá um peso na validação. A precisão da estimativa do erro melhora com o aumento de k, e no extremo onde k=N a técnica se torna o LOO, discutido abaixo. Esse processo de divisão, treinamento e validação do k-folds deve ser repetido várias vezes por causa da existência de uma variabilidade estatística na divisão dos conjuntos. Nesse trabalho foi utilizado k=10 e 5 repetições do k-folds.

B. LOO

A técnica LOO consiste em retirar uma amostra como amostra de validação e utilizar as restantes para treinamento, então repetir o processo para todas amostras e dividir o resultado pelo número de amostras. Enquanto isso possa parecer computacionalmente ineficiente, existe uma fórmula analítica para o cálculo do erro no caso dos modelos de projeção fixa:

$$\sigma_{LOO}^2 = \frac{1}{N} \mathbf{Y}^T \mathbf{P} (\text{diag}(\mathbf{P}))^{-2} \mathbf{P} \mathbf{Y} \quad (6)$$

com

$$\mathbf{P} = \mathbf{I}_N - \mathbf{H} \mathbf{A}^{-1} \mathbf{H}^T \quad (7)$$

onde

$$\mathbf{A} = \mathbf{H}^T \mathbf{H} + \lambda \mathbf{I}_P. \quad (8)$$

Repetindo esse processo para valores de λ , selecionamos o valor do hiper parâmetro que minimiza σ_{LOO}^2 .

C. Modelos de projeção fixa (ELMs e RBFs)

Para comparar os modelos, as amostras foram divididas em um conjunto de treinamento e um de validação, com a proporção exata de divisão variando entre os data sets, mas estando perto de 70% para treinamento.

Para o cálculo do erro sem regularização, as mesmas amostras do LOO foram utilizadas para treinamento, depois calculado o erro quadrático médio no conjunto de validação. Para o cálculo do erro LOO, um espaço de λ foi vasculhado para selecionar o melhor valor, e então o erro quadrático médio foi calculado no conjunto de validação utilizando o melhor λ .

Essa sequência de passos foi repetida para vários valores de neurônios, no caso da ELM, ou agrupamentos, no caso da RBF, para obter um escopo maior de análise da dependência de performance em relação a regularização e a forma de escolher o parâmetro de regularização.

D. Multilayer Perceptron

Sabemos que redes MLP são aproximadoras universais de neurônios [7], mas antes de uma análise dos efeitos da regularização é preciso determinar o número mínimo de neurônios para atingir uma boa performance durante o treinamento em todas amostras para cada dataset. Para isso, é só aumentar o número de neurônios e repetir o treinamento algumas vezes até o aumento dos neurônios não causar uma melhoria estatisticamente significativa na performance.

Depois, é preciso encontrar o λ que resulta no melhor erro de validação, no caso, a validação 10-fold foi repetida 5 vezes e depois feito a média. Para busca do λ foi feita uma busca com 10 pontos em um intervalo, com todos pontos exponencialmente espaçados.

E. Data sets

Como proposto pelo trabalho, foram utilizados 5 data sets para cada tipo de modelo (camada fixa ou treinada), sendo escolhidos 3 com tarefa de classificação e 2 com tarefa de regressão, mas para ELM e RBF foram utilizados dataset 1 ao 5, e para MLP os datasets 3 a 7. Abaixo está descrito a organização e distribuição de todos data sets.

1) *MNIST*: O data set *MNIST* [1] contém 70 mil imagens 28x28 em escala de cinza de dígitos individuais, as classes são quase igualmente distribuídas entre as 10 classes (dígitos de 0 a 9). Todas amostras são transformadas em um vetor de dimensão 784 e todos valores são divididos por 255 para ficarem normalizados entre 0 e 1, já que originalmente os valores são de 0 a 255.

Como o número de amostras e dimensões é muito grande, se todas amostras fossem ser utilizadas, o custo computacional seria muito grande para treinar em uma máquina gratuita do [Google Colab](#), então apenas uma parte dos dados pode ser utilizada. Além disso, para reduzir a complexidade do problema e o número de classes para 2, as amostras as amostras de classe '2' foram atribuídas como 1 e as amostras da classe '5' como -1, e todas outras amostras foram removidas, resultando em 13303 amostras.

2) *Mushroom*: O data set *Mushroom* [2] contém 8123 amostras com 3 atributos distribuídas entre duas classes, separando cogumelos entre 'comestível' e 'venenoso'. Todos atributos foram baixados em texto, tendo de ser convertidos para números, então uma classe foi arbitrariamente atribuída como -1 e a outra como 1, e os atributos foram atribuídos um número inteiro de acordo com a ordem que eles apareceram no dados.

3) *Statlog*: O data set *Statlog* [3] contém 690 amostras com 14 atributos, separadas entre 2 classes, e, como esses dados são clientes de um empresa de crédito, todos atributos tiveram seus significados removidos para preservar a anonimidade.

4) *Wine Quality*: O data set *Wine Quality* [4] tem como objetivo tentar prever a qualidade de um vinho baseado em atributos químicos, contém 6497 amostras com 11 atributos separadas entre vinhos vermelhos e vinhos brancos, mas foram utilizado apenas os dados do vinho vermelho com 1599 amostras.

5) *Real Estate Valuation*: O data set *Real estate valuation* [5] contém amostras 414 com atributos 14 de imóveis e o valor deles em 10000 Novo Dólar Taiwanês/Ping (1 Ping = 3,3 m²).

6) *Ionosphere*: O data set *Ionosphere* [8] contém amostras 351 com atributos 34 que representam valores coletados por 16 antenas e tenta classificar se há alguma estrutura na ionosfera.

7) *Sonar*: O data set *Sonar* [9] contém amostras 208 com atributos 60 que representam valores coletados por um sonar e tenta classificar o material que refletiu a onda recebida em rocha ou metal.

IV. RESULTADOS

O código utilizado para gerar os resultados abaixo estão disponíveis em um ambiente do [Google Colab](#) em Python.

Erro de Teste - MLP (Mean +- Sd)			
Dataset	Com Reg.	Sem Reg.	Sem Val.
Credit	0.78+-0.02	0.75+-0.04	0.87+-0.01
Wine	0.42+-0.04	0.41+-0.02	0.49+-0.01
Real	97+-15	93+-20	139+-2
Sonar	0.74+-0.08	0.75+-0.06	0.27+-0.01
Ion	0.86+-0.05	0.87+-0.05	0.64+-0.01

TABLE I: Erros de teste - MLP

A. Seleção de λ por LOO

Os resultados foram obtidos como discutido acima, sendo eles mostrados nas Figuras 1 até 5 (É recomendado uma ampliação do arquivo para melhor visualização das figuras).

B. Seleção de λ para redes MLPs

Primeiramente, os resultados da busca do número mínimo de neurônios necessários estão na Figura 6.

E na Figura 7 e na Tabela I estão apresentados os resultados da busca por lambda.

V. DISCUSSÃO

Pelos resultados da comparação entre com e sem regularização nos modelos de projeção fixa, pode ser observada a utilidade do LOO em todos data sets utilizando ELM e RBF, exceto na Figura 4 e 5, onde o resultado não é muito diferente. Mais especificamente, a regularização parece impedir o *overfitting*, e em poucos momentos resulta em *underfitting*, já que o valor de λ é escolhido por meio da técnica LOO.

Nas redes ELM, para data set é possível observar uma relação entre a dificuldade de obter um modelo bom e o melhor número de neurônios utilizado, e em todos casos essas redes se beneficiaram da regularização independente do número de neurônios, evitando *underfitting* e *overfitting*, mostrando um equilibrando entre o viés e a variância [6].

Já para RBFs, data sets 1, 4 e 5 não se beneficiaram muito de regularização, mas 2 e 3 precisaram de um grande número de agrupamentos e regularização para terem um melhor desempenho.

Já para as redes MLP, a regularização teve um resultado estatisticamente aos modelos não regularizados, isso se deve ao fato do número de neurônios utilizados para treinar foi no limitar do número mínimo de neurônios para "completar" a tarefa, o que faz a regularização não precisar ser aplicada, pois não há uma variância que pode ser retirada do modelo por meio da regularização. E também, os modelos sem regularização não realizaram um *overfitting* no conjunto de treinamento, já que a validação foi utilizada para decidir quando parar o treinamento do modelo.

VI. CONCLUSÃO

O uso da técnica LOO e k-folds para encontrar o hiperparâmetro de regularização mostrou resultados superior em relação aos modelos sem regularização, fazendo mesmo modelos com muitos elementos na camada fixa ganhassem um

grande aumento na performance, não obtido anteriormente sem regularização com modelos grandes ou pequenos.

REFERÊNCIAS

- [1] LeCun, Y., Cortes, C. and Burges, C.J.C. (1998) The MNIST Database of Handwritten Digits. New York, USA. <http://yann.lecun.com/exdb/mnist/>
- [2] Mushroom Data Set, <https://archive.ics.uci.edu/ml/datasets/Mushroom>
- [3] Statlog (Australian Credit Approval) Data Set , [https://archive.ics.uci.edu/ml/datasets/statlog+\(australian+credit+approval\)](https://archive.ics.uci.edu/ml/datasets/statlog+(australian+credit+approval))
- [4] Wine Quality Data Set, <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>
- [5] Real estate valuation data Set, <https://archive.ics.uci.edu/ml/datasets/Real+estate+valuation+data+set>
- [6] Geman, S., Bienenstock, E., Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. *Neural Computation*, 4(1), 1–58.
- [7] Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control Signal Systems* 2, 303–314 (1989). <https://doi.org/10.1007/BF02551274>
- [8] Ionosphere dataset, <https://archive.ics.uci.edu/ml/datasets/ionosphere>
- [9] Sonar dataset, [http://archive.ics.uci.edu/ml/datasets/connectionist+bench+\(sonar,+mines+vs.+rocks\)](http://archive.ics.uci.edu/ml/datasets/connectionist+bench+(sonar,+mines+vs.+rocks))

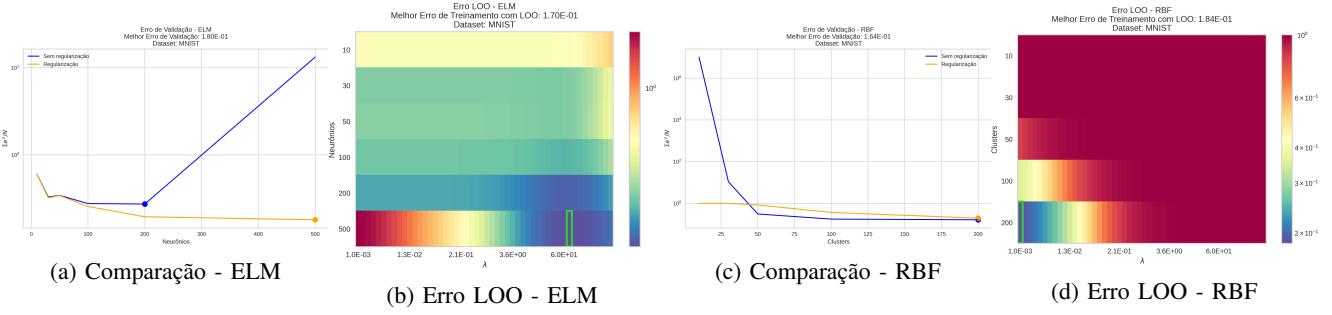


Fig. 1: Resultados do Dataset 'MNIST'

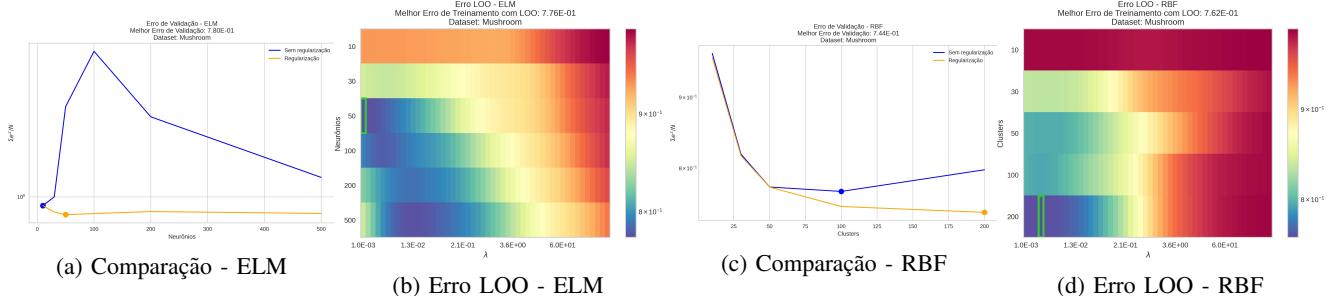


Fig. 2: Resultados do Dataset 'Mushroom'

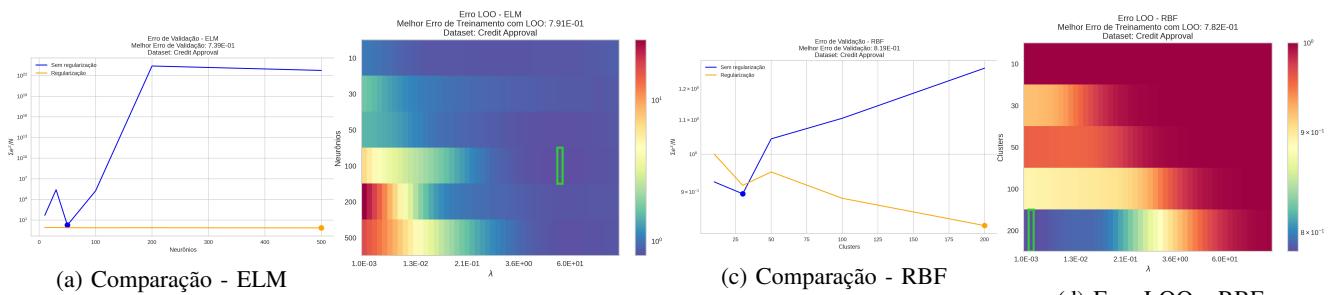


Fig. 3: Resultados do Dataset 'Credit Approval'

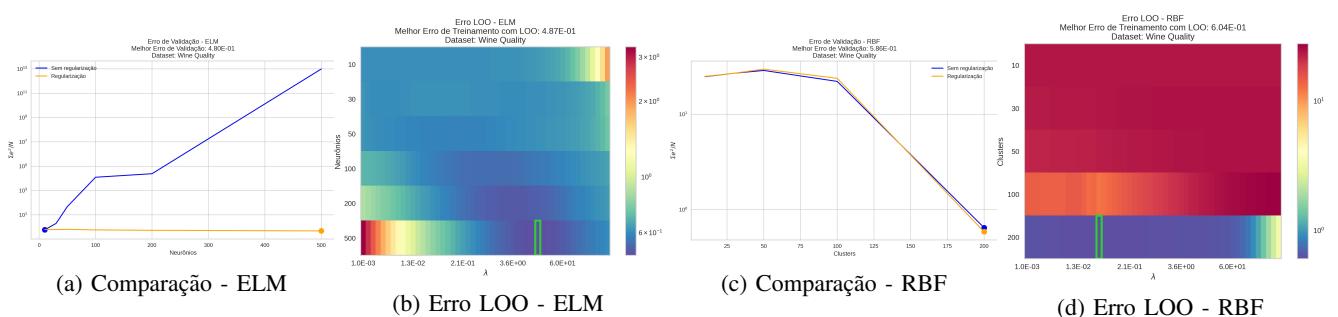


Fig. 4: Resultados do Dataset 'Wine Quality'

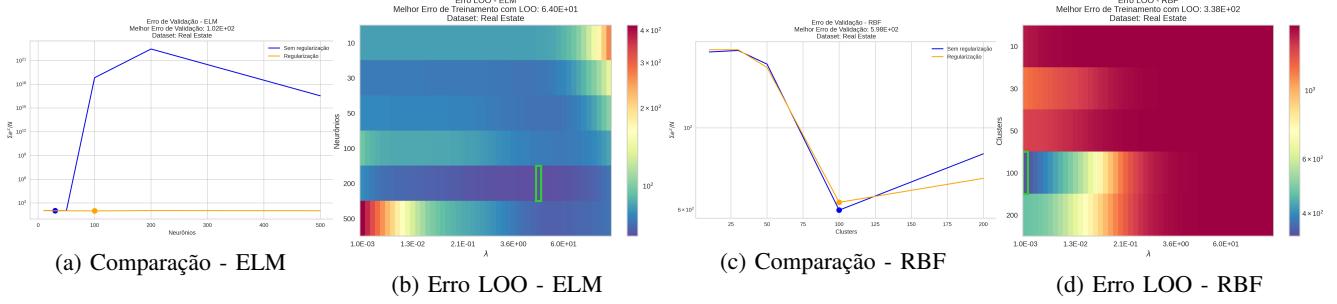


Fig. 5: Resultados do Dataset 'Real Estate'

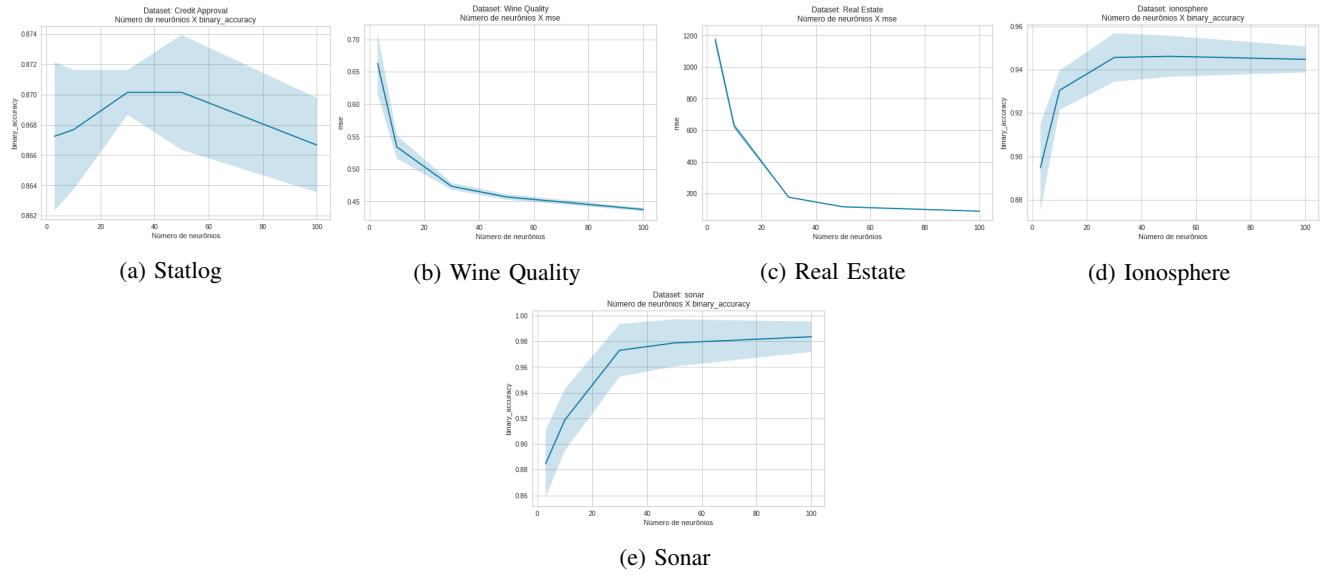


Fig. 6: Comparação do número de neurônios e erro de treinamento

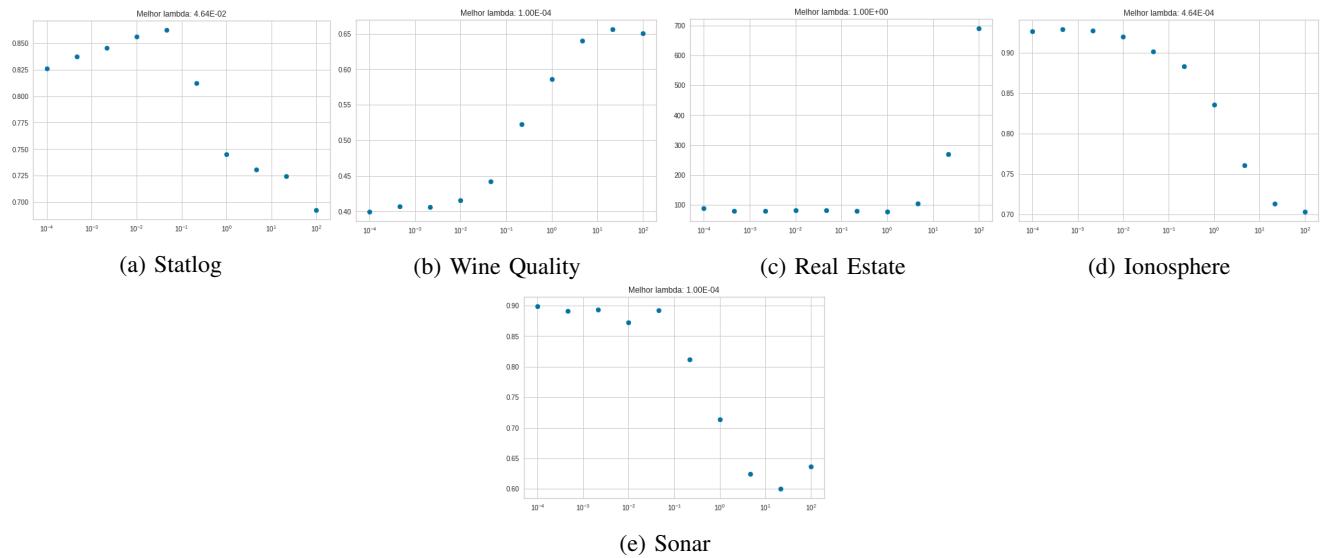


Fig. 7: Erro de teste - MLP