# Efeitos da Regularização em Redes Neurais Multicamadas do tipo RBF e ELM

André Costa Werneck

19 de Junho de 2022

## 1 Introdução

Redes neurais com múltiplas camadas muitas vezes atuam em problemas complexos e que apresentam dificuldades para serem resolvidos com redes de apenas uma camada. Tendo isso em mente, entende-se que a principal função das camadas escondidas é a de mapear as variáveis de entrada em um espaço intermediário que seja de fácil solução para a camada de saída. Essa solução, por sua vez, é considerada ótima para determinado problema quando ela é a que minimiza a função de custo aplicada na rede. Muitas vezes, essa função é, por exemplo, a soma dos erros quadráticos médios ou *MSE*, na sua sigla em inglês. Entretanto, uma solução de erro mínimo pode também resultar em *over-fitting*, ou sobre-ajuste, o que afeta bastante a capacidade de generalização do modelo e, assim, compromete seu desempenho. Logo, uma linha de estudos bastante relevante no mundo das redes neurais é como combater esse problema do sobre-ajuste.

Como citado em [2], uma forma importante de controlá-lo é limitar a magnitude dos pesos por meio de uma função de penalização, também conhecida como função de regularização do modelo. No caso do presente trabalho, estudou-se a *ridge regression* mostrada em [3] e definida pela seguinte equação:

$$J = \sum_{i=1}^{N}(y_i - yhat_i)^2 + \sum_{j=1}^{p} \lambda_j w_j^2 (1)$$

Em que N é o número de amostras, p é o número de funções radiais, no caso das redes RBF, e o número de neurônios na camada escondida, no caso das redes ELM, e lambda é o termo de penalização do peso wj. Logo, a partir da equação acima, observa-se que quando lambda igual 0, tem-se a equação comum de somatório dos erros quadráticos de um modelo e que quando lambda difere de zero, quanto maior o peso, maior é o valor da multiplicação do fator de regularização com respectivo peso e, assim, diz-se que o modelo "penaliza"pesos de maior módulo. Dessa forma, o mais relevante aqui é compreender que, da maneira descrita anteriormente, ao atribuir uma penalidade ao modelo, a regularização faz com que o erro seja maior, suavizando a resposta da rede e, principalmente, reduzindo os efeitos de *overfitting*.

## 2 Revisão de Trabalhos Correlatos

Como citado acima, o estudo de técnicas de regularização em *machine learning* é vasto e já existem diversos trabalhos publicados que auxiliam no entendimento do problema.

Na área supracitada, provavelmente, o principal problema é o de encontrar hiper-parâmetros (no nosso caso apenas o Lambda) que representem a melhor relação entre o viés e a variância. Nesse sentido, [4] mostra que esse problema não é trivial e são explicadas inúmeras soluções já conhecidas para buscar um bom parâmetro Lambda em casos de *Ridge Regression*, como o Método da Covariância e o *Eigen Method*, por exemplo. Assemelhando-se neles, em extração de *features* dos dados e usando *cross-validation*, os autores estenderam a literatura para otimizar a escolha do hiper-parâmetro em questão em grandes *datasets*. Além disso, eles incluíram uma análise interessante em relação à complexidade computacional do algoritmo desenvolvido e o compararam com alguns dos métodos mais comuns praticados na área, como o *Leave-One Out Cross-Validation*, explicitando seu alto custo de processamento, ainda mais elevado em grandes conjuntos de dados. Entretanto, mesmo conseguindo obter um método leve em termos de custo computacional para grandes bases de dados, não foi apresentada nenhuma solução que determinasse lambda sem usar *cross-validation* e, ademais, sua complexidade de implementação é consideravelmente elevada, o que justificou a não implementação do método no presente trabalho.

Complementar ao apresentado em [4], os autores em [5] ensinam de forma clara o que é a fronteira de Pareto e propõe uma solução para o Problema de Seleção de Modelos a partir da minimização da autocorrelação de resíduos inseridos no conjunto de Pareto. Por mais que a solução apresentada por [5] pareça inovadora e robusta tanto em termos de resultado quanto em termos de custo computacional e apresente uma ideia bastante semelhante ao que está sendo proposto neste trabalho, foi utilizada por [5] uma rede MLP, o que implicaria adaptar a solução proposta para os modelos de RBFs e ELMs e isso não fez parte do escopo deste estudo.

## 3 Metodologia

Visando conhecer os efeitos da regularização nos modelos de aprendizado de máquina, foram implementados duas redes: uma RBF e uma ELM, primeiramente, sem nenhum tipo de técnica de decaimento de pesos. Como mostrado em [1], logo após, adaptou-se a estrutura da rede para incluir a regularização através da inclusão da matriz A, após a obtenção da matriz de mapeamento para a camada intermediária, H, e, com elas, calcular o vetor de pesos, da seguinte forma:

$$\mathbf{A} = (\mathbf{H}^T H + \lambda I)(2)$$

$$\mathbf{w} = \mathbf{inv(A)}\mathbf{H}^T y(3)$$

Feito isso, tem-se prontos os modelos regularizáveis, nos quais é possível realizar algum controle sobre o *overfitting*, com já elucidado acima. Logo após, levantou-se 3 bases de dados diferentes (Sinc, Breast Cancer,Iris) para testar as redes ELM e RBF

com e sem regularização, assim como para validar se o aprendido na teoria aconteceria na prática. Para isso, entretanto, era preciso encontrar uma solução para o problema de selecionar um modelo com o melhor hiper-parâmetro Lambda possível, como visto nos trabalhos correlatos. Para resolver tal questão, escolheu-se utilizar o método *Leave One Out Cross Validation (LOOCV)* e avaliar o modelo com o erro quadrático médio do respectivo método de validação cruzada, o qual é definido como segue:

$$\sigma^2_{LOO} = (1/N) y^T \mathbf{P(diag(P))}^{-2} Py \quad (4)$$

No LOOCV, define-se uma lista de valores de lambda a serem testados e se escolhe, arbitrariamente, uma amostra do conjunto de treinamento para servir como amostra de validação. Após isso, separam-se os dados e treina-se todas as outras amostras com um dos valores de Lambda já previamente definidos. Com o treinamento feito, valida-se o modelo e calcula-se o erro de LOOCV para aquele respectivo valor de lambda. Logo depois, muda-se o valor de lambda, muda-se a amostra de validação e se repete o algoritmo descrito acima. Dessa forma, após treinar e validar o modelo com todos os valores de lambda desejados e passando por todo o conjunto de treinamento, obtém-se uma lista de valores de erro de LOOCV. Com isso em mãos, analisam-se os erros e o modelo com o valor de lambda que minimizou o Sigma de LOOCV é escolhido como ideal. Assim, com um modelo já definido, treinou-se todas as amostras de treino novamente, validou-se com uma parte do conjunto de dados nunca testada e já separada de antemão e testou-se com outro conjunto de dados da mesma base. Além disso, foram realizados treino, validação e teste na mesma base de dados, mas, dessa vez, para um modelo sem regularização.

Dessa forma, o objeto de análise do presente estudo foi a comparação dos resultados obtidos com os diferentes modelos: regularizados e não-regularizados.

# 4 Resultados

Escolheu-se começar de maneira simples e realizar um estudo reproduzindo um exemplo enunciado em [2]. Resolveu-se aproximar a função ***Sinc*** e os resultados foram os vistos nas figuras abaixo. Vale ressaltar que, na primeira, os valores em verde são da função geradora, os valores em azul são sem regularização e os em amarelo são com regularização.

A segunda base escolhida foi a já conhecida base nativa do R de dados de Cancer de mama, ***Breast Cancer***:

| $\lambda$ | $\sigma^2_{LOO}$ |
|---|---|
| 0 | 0.15 |
| 0.2 | 0.107 |
| 0.4 | 0.111 |
| 0.6 | 0.113 |
| 0.8 | 0.117 |
| 1.0 | 0.118 |

| Acuracia C/Reg. | Acuracia S/Reg. |
|---|---|
| 0.98364 | 0.97750 |

A terceira base escolhida foi a ***Iris***, já conhecida e também nativa do R.

| $\lambda$ | $\sigma^2_{LOO}$ |
|---|---|
| 0 | 0.3963 |
| 0.2 | 0.3952 |
| 0.4 | 0.3948 |
| 0.6 | 0.3951 |
| 0.8 | 0.03957 |
| 1.0 | 0.3964 |

| Acuracia C/Reg. | Acuracia S/Reg. |
|---|---|
| 1.0 | 1.0 |

# 5 Discussões

O experimento com a função Sinc foi bastante interessante, já que, ficou muito clara a metodologia explicada acima para seleção do Lambda ótimo, com o gráfico da segunda figura. Além disso, o efeito positivo da regularização na aproximação da função é evidente, visto que, sem regularização, existem inúmeros pontos fora de um lugar pertencente à curva da função, representando os ruídos. E, já com a regularização, a resposta é mais suave, com menos ruídos e, portanto, mais semelhante à função geradora.

Para a ***Breast Cancer*** fica claro que o melhor lambda para o modelo é o de valor igual a 0.2, uma vez que os erros de LOOCV são crescentes antes e após ele. Além disso, vale dizer que se observou um crescimento na acurácia representando uma melhora significativa na performance do classificador. Entretanto, como esse é um problema simples de se resolver, mesmo sem regularização presente, é possível observar uma acurácia próxima a 100%, o que já configura uma boa solução.

Já para a base da ***Iris***, por mais que tenha sido possível observar também uma convergência do erro de LOOCV até um ponto mínimo e, com isso, a seleção de um Lambda ótimo, a acurácia não mudou e continuou em 100% para ambos os testes, com e sem regularização. Isso se deve ao fato de que, graças à simplicidade do modelo, a solução ótima do problema é facilmente encontrada e, nesse caso, a regularização se faz desnecessária.

# 6 Conclusões

Após o estudo tanto teórico quanto prático das técnicas de regularização e de seus efeitos, fica claro que elas afetam o desempenho dos modelos aos quais são aplicadas.

Entretanto, nem sempre elas são realmente benéficas para os mesmos. Ficou evidente que, em problemas mais complexos em que há a tendência de um superdimensionamento da rede e, assim, há também maior chance de ocorrer um *overfitting*, regularizar o aprendizado é de extrema relevância. Porém, em casos contrários, mais simples e de fácil resolução, a presença de técnicas como a de decaimento de pesos torna-se desnecessária e pode até prejudicar um pouco, uma vez que acrescenta custos computacionais aos respectivos problemas nos quais são utilizadas, já que álgebra matricial e técnicas de cross-validation necessárias para determinação dos hiperparâmetros podem ser bem custosas em algumas situações.

Ademais, é importante dizer que, sem uma escolha bem feita de Lambda, por exemplo, usar regularização pode ter uma relação direta com a queda da acurácia do modelo o que, nesses casos, representa o efeito inverso ao que se deseja obter utilizando tais artificios.

É importante ressaltar também, que o uso do erro de LOOCV re-

presentou um grande aprendizado e é uma técnica clássica e poderosa para avaliação do desempenho dos modelos considerando os *trade offs* necessários no problema do viés e da variância.

Por fim, conclui-se que as técnicas de regularização são de suma importância se usadas corretamente e para os problemas apropriados. Dessa forma, elas não só representam ferramentas de suavização de sobre-ajustes, mas também facilitam a obtenção de soluções mais fielmente replicáveis, uma vez que reduzem o conjunto possível de soluções e também corroboram para que os pesos não fiquem presos em mínimos locais não ótimos.

# Referências


[1] Antônio de Pádua Braga. *Notas de Aula de Redes Neurais Artificiais e de Reconhecimento de Padrões*. Escola de Engenharia UFMG, 2018.

[2] Antônio de Pádua Braga. *Aprendendo com Exemplos : Principios de Redes Neurais Artificiais e de Reconhecimento de Padrões*. Escola de Engenharia UFMG, 2021.

[3] Arthur E Hoerl and Robert W Kennard. Ridge regression : Biased estimation for nonorthogonal problems. *Technometrics, 12(1):55-67*, 1970.

[4] Joni Dambre David Verstraeten Benjamin Schrauwen Pieter Buteneers, Ken Caluwaerts. Optimized parameter search for large datasets of the regularization parameter and feature selection for ridge regression. *Springer Science+Business Media New York*, 2013.

[5] Antônio de Pádua Braga Talles Medeiros, Ricardo Takahashi. A new decision strategy in multi-objective training of the artificial neural networks. *ESANN*, 2007.

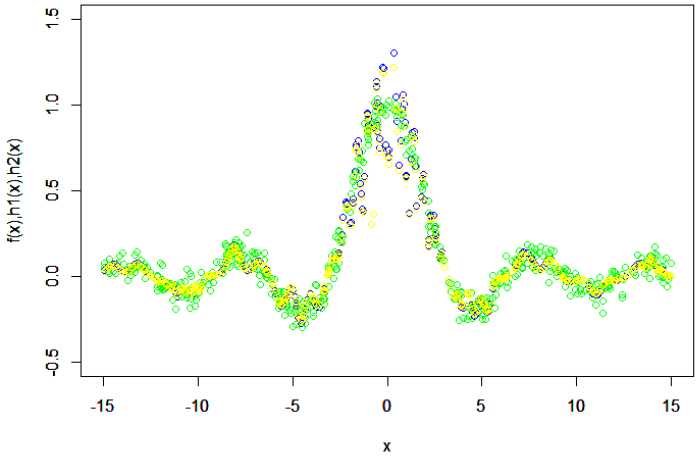


Figura 1: Aproximações Sinc

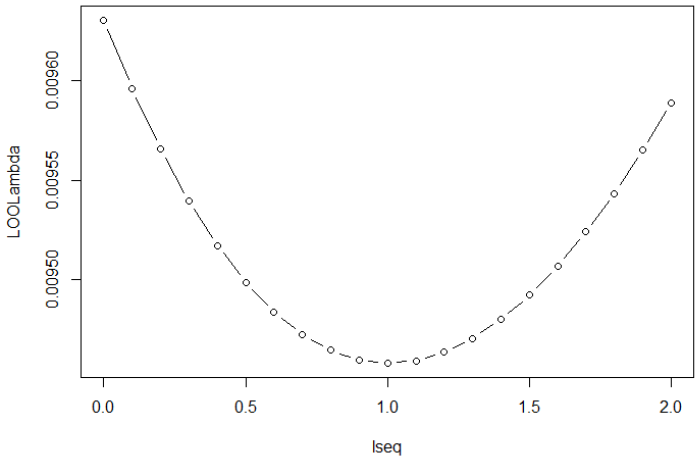


Figura 2: Erro de LOOCV Sinc