

# Regularização de redes neurais artificiais

João Pedro Miranda Marques - 2017050495

6 de julho de 2022

## 1 Introdução

A Regularização está relacionada a um dos dois fundamentais erros estatísticos em modelos científicos: overfitting e underfitting, como afirma McElreath em *Statistical Rethinking*[2].

Modelos com Underfitting possuem baixa capacidade de relacionar os dados de entrada e saída gerando assim uma elevada taxa de erro tanto nos dados de treinamento quanto de validação e teste.

Por outro lado o Overfitting, segundo Douglas M. Hawkins em *The Problem of Overfitting*[1], é o uso de modelos ou procedimentos que violam a parcimônia, ou seja, que incluem mais termos do que o necessário ou usam abordagens mais complicadas do que o necessário.[1] Dessa maneira o overfitting apresentam uma aproximação muito detalhada com relação aos dados de treinamento, absorvendo assim informações de ruído que não pertencem à função geradora do problema. Como resultado apresentam uma ótima acurácia com relação aos dados de treinamento. Porém ao utilizar novos dados há prejuízos devido esse deslocamento da função geradora do problema.

Redes Neurais solucionam problemas de maneira multiobjetivo<sup>1</sup>, ou seja, costumam ter não só uma solução ótima, mas um conjunto de soluções ótimas, denominado conjunto Pareto-Ótimas(PO). Esse conjunto PO é representado na Figura 1 pela região decrescente da curva em vermelho, nessa região há um trade-off entre erro de generalização e complexidade do modelo.

Vemos também que a curva em vermelho apresenta um vale. Nesse ponto encontramos a menor taxa de erro de generalização para o modelo utilizado. Vemos que a medida que aumenta a complexidade do modelo temos uma absorção do ruído de treinamento ocasionando o overfitting.

A regularização portanto consiste em métodos para reduzir os efeitos de overfitting e aproximar nossa solução do conjunto Pareto-Ótimo.

Este trabalho tem o objetivo de estudar os métodos de regularização implementados em redes ELM<sup>2</sup>, RBF<sup>3</sup> e MLP<sup>4</sup> e avaliar seu desempenho.

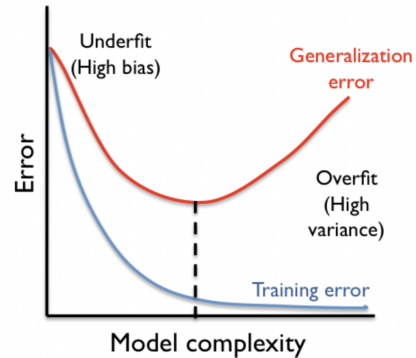


Figura 1: Conjunto de soluções para o problema.

## 2 Revisão de trabalhos correlados

O algoritmo de regularização é encontrado em diversas áreas de estudo, porém com nomes diferentes. Na estatística é conhecido como *Ridge regression*, em otimização de sistemas é *Regularização de Tikhonov* e em aprendizado de máquina como *weight decay*. O conceito de regularização abrange também outros algoritmos de otimização. Dentre eles o *Pruning*<sup>5</sup>, OBD<sup>6</sup>, OBS<sup>7</sup> dentro outros.

Na tese de pós-graduação do autor Medeiros[3] há a proposta de um novo método de poda no qual ele nomeou de CAPE. Esse método é implementado em redes MLP<sup>8</sup> e tem por objetivo reduzir o custo computacional da inversão da matriz Hessiana, cálculo esse muito utilizado em outros algoritmos. Em seguida o autor implementa o algoritmo em um caso real de identificação de falhas em motores de indução para avaliar e comparar a performance em relação ao algoritmo OBS.

## 3 Metodologia

O livro *Redes neurais artificiais: teoria e aplicações*[4] explica o algoritmo de regularização weight decay e au-

<sup>1</sup>Problemas multiobjetivo: Problemas com mais de um objetivo.

<sup>2</sup>ELM: Extreme Learning Machines.

<sup>3</sup>RBF: Radial Basis Function.

<sup>4</sup>MLP: Multilayer Perceptron.

<sup>5</sup>Pruning (poda) algoritmo muito utilizado em Arvore de decisao.

<sup>6</sup>OBD: Optimal Brain Damage.

<sup>7</sup>OBS: Optimal Brain Surgeon.

<sup>8</sup>MLP: Multilayer Perceptron.

xilia sua implementação em redes RBF. A metodologia utilizada define uma variável Lâmbida que representa o termo de penalização do vetor de pesos  $W$ . O efeito do Lâmbida é deslocar a solução do mínimo local da região erro de treinamento com o objetivo de reduzir a absorção de ruído e consequentemente o overfitting.

A regularização foi implementada no algoritmo de treinamento da rede RBF na linguagem python. O código foi feito traduzindo os exemplos em R disponíveis no livro texto e pode ser encontrado no github<sup>9</sup> desse trabalho.

$$\mathbf{w} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I}_p)^{-1} \mathbf{H}^T \mathbf{y}$$

No algoritmo o vetor de pesos  $W$  é influenciado não só pela matriz  $H$  que armazena informações do modelo, mas também pela matriz gerada pelo termo de regularização. Dessa forma, no algoritmo foi adicionado o cálculo da matriz de Variância  $A$ , que corresponde ao primeiro termo do cálculo dos pesos  $W$ .

$$\mathbf{A}^{-1} = \mathbf{H}^T \mathbf{H} + \lambda \mathbf{I}_p$$

O cálculo da Matriz de Projeção  $P$ , que será utilizada para cálculo da função Custo  $J$  do modelo.

$$\mathbf{P} = \mathbf{I}_N - \mathbf{H} \mathbf{A}^{-1} \mathbf{H}^T$$

Diversos experimentos foram realizados utilizando esse algoritmo de redes RBF para analisar o efeito da regularização. Na próxima seção veremos os resultados.

Para os experimentos utilizando redes Multilayer Perceptron MLP, foram utilizados as bibliotecas prontas para python scikitlearn[5].

## 4 Resultados

### Experimento 1

Utilizando a rede RBF com regularização foi realizado um problema de classificação com os pacotes de dados make circles e make moons disponíveis no repositório sklearn.<sup>10</sup>

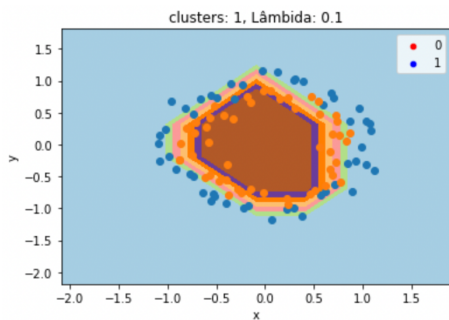


Figura 2: RBF: 1 cluster e regularização = 0.1

<sup>9</sup>LINK DO GIT.

<sup>10</sup><https://scikit-learn.org/stable/>.

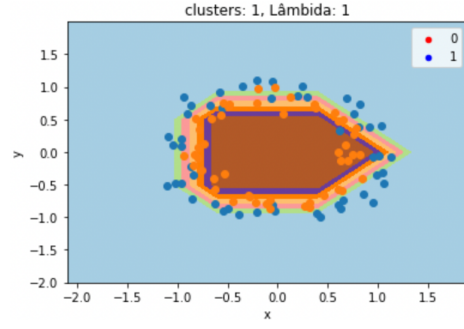


Figura 3: RBF: 1 cluster e regularização = 1

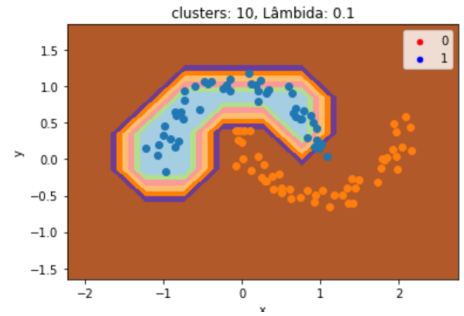


Figura 4: RBF: 10 cluster e regularização = 0.1

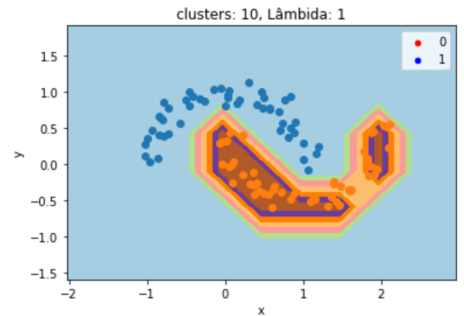


Figura 5: RBF: 10 cluster e regularização = 1

Nesse experimento a quantidade de clusters foi escolhida por meio de testes de forma a gerar o melhor resultado para cada base de dados selecionada. Como o objetivo é avaliar o efeito da regularização, a quantidade de clusters se manteu fixo.

### Experimento 2

Esse experimento foi realizado utilizando a base de dados Iris. Nessa base de dados há a classificação de três tipos de flores de acordo com o comprimento e largura da sépala da flor. Na figura 6 podemos ver a distribuição dos dados para as três classificações.

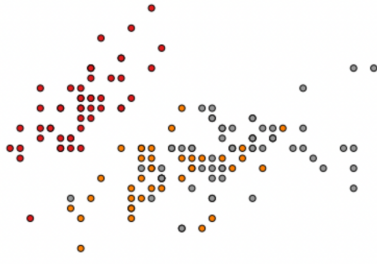


Figura 6: Distribuição dos dados

Uma rede RBF foi treinada com 3 clusters e valores de  $\lambda$  variando de 0.1 a 1. Para cada valor de  $\lambda$  foi calculado o somatório do erro quadrático de predição e a norma quadrática dos pesos de treinamento  $w$ .

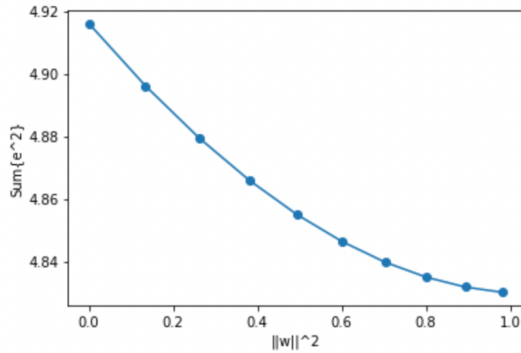


Figura 7: Conjunto de soluções regularizadas

Podemos ver a curva de soluções Pareto-Ótimas na figura 7

### Experimento 3

A base de dados utilizada nesse experimento foi a Breast Cancer disponibilizada na biblioteca *sklearn*. Essa base de dados tem por finalidade prever se um paciente apresenta sinais de câncer de mama de acordo com o resultado de diversos exames.

Para esse experimento foi analisado a acurácia média de previsão do algoritmo para diversos valores de  $\lambda$  na regularização de uma rede neural RBF. O resultado está explicitado na figura 8

```
Acurácia média com Lâmbda 0.1: 0.6573099415204678 +/- 0.022005716634183384
Acurácia média com Lâmbda 0.3: 0.6690058479532163 +/- 0.017189401703741586
Acurácia média com Lâmbda 0.6: 0.6701754385964912 +/- 0.033817151566785865
Acurácia média com Lâmbda 1: 0.6923976608187135 +/- 0.036874579579963616
```

Figura 8: Acurácia da rede RBF regularizada no dataset Breast Cancer

### Experimento 4

Por fim, foi implementado o mesmo teste realizado no experimento 2, porém com o dataset Breast Cancer.

<sup>11</sup>Multilayer Perceptron.

<sup>12</sup>Multilayer Perceptron.

cer. Podemos ver na figura 10 que a curva de soluções não apresentou uma característica de decaimento semelhante ao da figura 7.

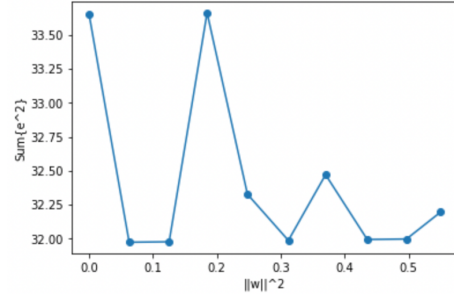


Figura 9: Soluções de uma rede RBF regularizada com dataset Breast Cancer

### Experimento 5 - MLP

Neste experimento foi utilizado uma rede de perceptrons em múltiplas camadas MLP<sup>11</sup> para avaliar o efeito da regularização. Utilizando função *MLPClassifier* da biblioteca *sklearn*[5] podemos selecionar o parâmetro de regularização  $\lambda$  referente ao processo de regularização Ridge (L2). Podemos ver o efeito desse parâmetro na figura 10.

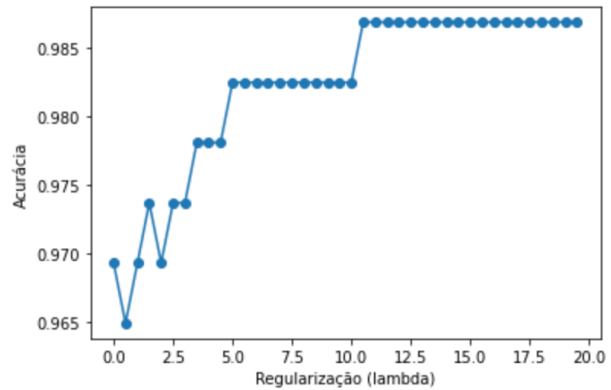


Figura 10: Soluções de uma rede MLP regularizada com dataset Breast Cancer

### Experimento 6 - MLP

Utilizado novamente uma rede MLP<sup>12</sup> nesse exemplo foi feito com a base de dados digits. Nele calculamos a acurácia para regularização L1 - Ridge com fatores de penalização 0.01, 0.1 e 0.9. Para cada um deles vemos também a matriz de confusão de cada modelo predito. Podemos ver o efeito desse parâmetro na figura 11.

```

Data preprocessing with StandardScaler()

lambda = 0.01
MLP Accuracy: 97.41%
MLP Training set score: 100.00%
MLP Testing set score: 97.41%
-----

lambda = 0.1
MLP Accuracy: 97.04%
MLP Training set score: 99.92%
MLP Testing set score: 97.04%
-----

lambda = 0.9
MLP Accuracy: 95.93%
MLP Training set score: 98.17%
MLP Testing set score: 95.93%

```

Figura 11: Soluções de uma rede MLP regularizada com dataset Digits

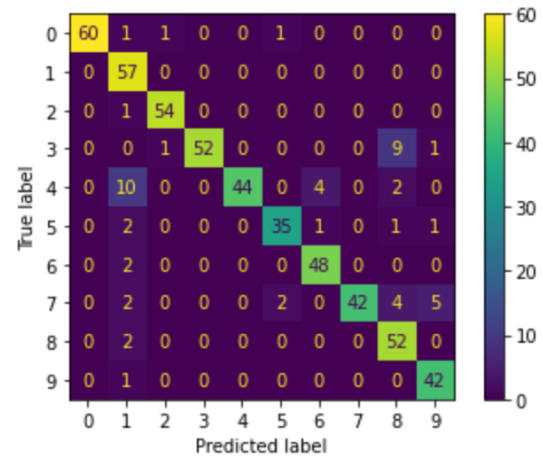


Figura 14: Matriz de confusão - lambda = 0.9

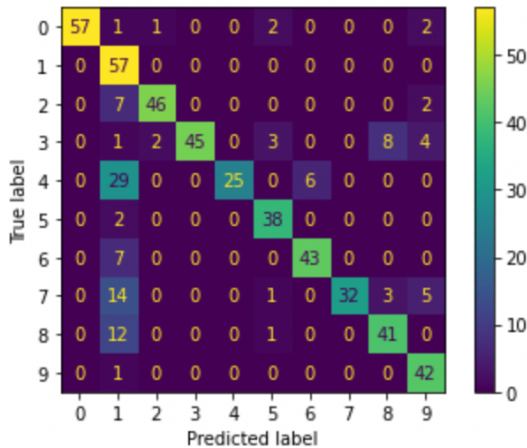


Figura 12: Matriz de confusão - lambda = 0.01

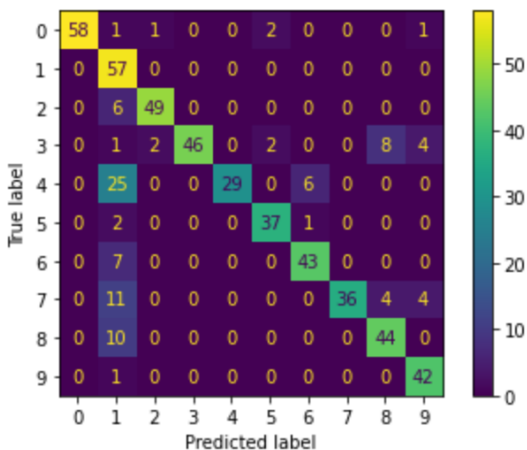


Figura 13: Matriz de confusão - lambda = 0.1

## 5 Discussões

### Experimento 1

Nesse experimento podemos ver visualmente que o efeito da regularização não afetou muito os resultados de classificação obtidos, o que é um resultado desejado visto que conseguimos reduzir a complexidade do modelo e ainda obter a resposta dentro do esperado.

### Experimento 2

Nesse experimento o foco foi na curva de soluções pareto-ótimas do problema. Qualquer solução escolhida nessa curva é um ponto ótimo entre o tradeoff de erro de predição e complexidade do modelo. Soluções acima dessa curva são consideradas soluções não otimizadas e as soluções abaixo da curva são não tangíveis ao problema e modelo utilizados. (considerando a quantidade de clusters da rede RBF como característica do modelo escolhido).

### Experimento 3 e 4

Nos experimentos utilizando o dataset do breast cancer os resultados com relação a regularização não foram conforme o esperado. No experimento 3 a acurácia subiu conforme aumentou o valor de lambda. Esse resultado é o oposto do esperado visto que estamos conseguindo melhorar os dois objetivos, portanto ainda não chegamos na região ótima de soluções. Já no experimento 4 podemos ver que a curva não tem o formato desejado igual a vista no experimento 2. Dessa forma, podemos deduzir que o modelo e a quantidade de clusters selecionada não conseguiu aproximar de um conjunto de soluções pareto-ótimas.

### Experimento 5

Utilizando uma rede MLP e a base de dados Breast Cancer vemos novamente um comportamento não esperado, onde quando aumentamos do fator de regularização

apresentamos uma acurácia ainda maior. Isso pode ser devido ao overfitting do modelo com relação ao conjunto de dados de treinamento, o que poderia estar prejudicando a acurácia com relação ao conjunto de dados de testes. Essa hipótese também é reforçada pelo fato de a acurácia ter melhorado apenas de 0.97 para 0.99, o que não foi uma brusca mudança mostrando que com a regularização conseguimos suavizar o ruído de erro no conjunto de treinamento.

## Experimento 6

Utilizando novamente uma rede MLP, porém dessa vez com a base de dados Digits podemos ver que aumentando o fator de regularização temos uma leve redução na acurácia de testes. De 0.974 para 0.959, o que nos mostra que a simplificação no modelo devido a regularização piorou a acurácia porém pode continuar sendo um resultado positivo pois o modelo por ser mais simples apresenta melhor performance de predição. Podemos ver essa simplificação de forma visual ao avaliar a matriz de confusão de cada modelo. Vemos que com o maior  $\lambda$  temos uma matriz mais diagonalizada, representando um modelo com pesos menores.

## 6 Conclusões

O Trabalho tem como objetivo fazer um estudo, compreensão e testes de regularização de redes neurais artificiais. Durante o desenvolvimento do trabalho foram realizadas várias pesquisas sobre o funcionamento e implementação desse método. Fiz uma busca de pacotes de modelos prontos de ELM e RBF com regularização em python mas nenhuma chegou a gerar um experimento interessante. Muitos dos exemplos encontrados eram realizados utilizando outros métodos de reconhecimento de padrões, como SVM e Regressão Linear para estudar efeitos de regularização. Por esse motivo, o trabalho foi desenvolvido implementando a metodologia de weight decay explicitada no Livro texto[4] na minha implementação do algoritmo de RBF na linguagem python.

Conseguimos chegar a resultados esperados em algumas bases de dados e em resultados não desejados no dataset Breast Cancer. Pude fazer uma avaliação do porque o resultado divergiu.

Por fim, com esse trabalho temos uma maior compreensão sobre o tema de regularização e otimização de modelos de redes neurais artificiais.

## Referências

- [1] Douglas M. Hawkins. “The Problem of Overfitting”. Em: *Journal of Chemical Information and Computer Sciences* 44.1 (2004). PMID: 14741005, pp. 1–12. DOI: [10.1021/ci0342472](https://doi.org/10.1021/ci0342472). eprint: <https://doi.org/10.1021/ci0342472>. URL: <https://doi.org/10.1021/ci0342472>.
- [2] R. McElreath. *Statistical Rethinking: A Bayesian Course with Examples in R and Stan*. Chapman & Hall/CRC Texts in Statistical Science. CRC Press, 2016. ISBN: 9781482253467. URL: <https://books.google.com.br/books?id=mDo0CwAAQBAJ>.
- [3] Cláudio Marques de Sá Medeiros. “A contribution to the problem of selection of neural models using the beginning of maximum correlation of the errors.” Em: *Tese (Doutorado em Engenharia de Teleinformática)-Centro de Tecnologia, Universidade Federal do Ceará, Fortaleza* (2008). URL: <http://www.repositorio.ufc.br/handle/riufc/16117>.
- [4] A. de Pádua Braga, A.C.P. de Leon Ferreira Carvalho e T.B. Ludermit. *Redes neurais artificiais: teoria e aplicações*. Livros Técnicos e Científicos, 2000. ISBN: 9788521612186. URL: <https://books.google.com.br/books?id=cUgEaAEACAAJ>.
- [5] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. Em: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.