UNIVERSIDADE FEDERAL DE MINAS GERAIS
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Alex Damiany Assis

REAMOSTRAGEM LOCAL BASEADA EM INFORMAÇÃO ESTRUTURAL DOS DADOS COM REGULARIZAÇÃO DE REDES NEURAIS ARTIFICIAIS

Belo Horizonte
2022

Alex Damiany Assis

REAMOSTRAGEM LOCAL BASEADA EM INFORMAÇÃO ESTRUTURAL DOS DADOS COM REGULARIZAÇÃO DE REDES NEURAIS ARTIFICIAIS

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito parcial para obtenção do Título de Doutor em Engenharia Elétrica.

Orientador: Antônio de Pádua Braga

Coorientador: Luiz Carlos Bambirra Torres

Belo Horizonte
2022

Assis, Alex Damiany.
A848r Reamostragem local baseada em informação estrutural dos dados com regularização de redes neurais artificiais [recurso eletrônico] / Alex Damiany Assis.- 2022.
1 recurso online (110 f. : il., color.) : pdf.

Orientador: Antônio de Pádua Braga.
Coorientador: Luiz Carlos Bambirra Torres.

Tese (doutorado) - Universidade Federal de Minas Gerais, Escola de Engenharia.

Apêndices: f. 91-110.
Bibliografia: f. 85-90.

Exigências do sistema: Adobe Acrobat Reader.

1. Engenharia elétrica - Teses. 2. Redes Neurais (Computação) – Teses. I. Braga, Antônio de Pádua. II. Torres, Luiz Carlos Bambirra. III. Universidade Federal de Minas Gerais. Escola de Engenharia. IV. Título.

CDU: 621.3(043)

Ficha catalográfica elaborada pela Bibliotecária Letícia Alves Vieira - CRB-6/2337
Biblioteca Prof. Mário Werneck - Escola de Engenharia da UFMG

"Reamostragem Local Baseada Em Informação Estrutural dos Dados Com Regularização de Redes Neurais Artificiais"

Alex Damiany Assis

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 03 de janeiro de 2022.

Por:

Prof. Dr. Antônio de Pádua Braga
DELT (UFMG) - Orientador

Prof. Dr. Luiz Carlos Bambirra Torres
Departamento de Computação e Sistemas (UFOP) - Coorientador

Prof. Dr. Eduardo Mazoni Andrade Marçal Mendes
DELT (UFMG)

Prof. Dr. João Pedro Braga
Programa de Pós-Graduação em Química (UFMG)

Prof. Dr. Haroldo Fraga de Campos Velho
Coordenação de Pesquisa Aplicada e Desenvolvimento Tecnológico (INPE)

Prof. Dr. Elcio Hideiti Shiguemori
Departamento de Ciência e Tecnologia Aeroespacial (DCTA-IEAv)

Dedico esta tese a minha esposa Fernanda, as minhas filhas Maria Luisa, Giovana e Mariana, aos meus pais e a minha irmã.

Agradecimentos

À Deus pelo dom da vida e pelo sustento na caminhada hoje e sempre.

À Fernanda pelo apoio, paciência, companheirismo e ajuda durante todos esses anos. As minhas filhas Maria Luisa, Giovana e Mariana por trazerem calma, paz e conforto em momentos de tribulação.

Aos meus pais, João e Benedita, sempre presentes e nunca mediram esforços para me apoiar na minha caminhada. A minha irmã Aline pelo apoio e estar sempre presente.

Ao Prof. Antônio de Pádua Braga, pela oportunidade de participar da família LITC, pela dedicação, paciência e sabedoria que teve comigo na realização desta tese. Os ensinamentos foram além da academia pelo exemplo de pessoa e educador. Sou eternamente grato.

Ao meu coorientador Prof. Luiz Carlos Bambirra Torres, pela amizade, dedicação, orientação e conselhos na condução deste trabalho.

Aos amigos do LITC, pelas boas conversas.

À Vera por cuidar da organização e limpeza do LITC.

Ao departamento de Economia da UFJF/GV por conceder o afastamento para realização do doutorado.

À Pró-Reitoria de Gestão de Pessoas UFJF pelo apoio e suporte financeiro.

Resumo

A capacidade de aprendizado de uma rede neural artificial depende das restrições impostas ao seu espaço de soluções que podem ser determinadas pelo número de parâmetros do modelo ou por outras formas de restrições de busca neste espaço. A complexidade da rede neural pode ser controlada pela decomposição da esperança do erro quadrático em dois termos que representam o viés e a variância da família de modelos. Uma técnica utilizada para controlar o *trade-off* entre o viés e a variância é a regularização, que controla a variância pela modificação da função de erro com adição de um termo de penalização. A proposta deste trabalho é definir um classificador de margem larga baseado na adição de amostras sintéticas no conjunto de treinamento em seu espaço de características. A abordagem proposta é baseada em um modelo de adição de ruídos no treinamento e das informações estruturais dos dados. Os experimentos foram realizados para comparar o modelo proposto denominado como *Regularization with Noise of Extreme Learning Machine* (RN-ELM) em relação a *Extreme Learning Machine* (ELM) padrão e o modelo de *Extreme Learning Machine* com regularização (ELM-REG). Os resultados mostram a capacidade de reduzir a norma e suavizar a superfície de separação dos modelos RN-ELM e ELM-REG. Na avaliação estatística da acurácia média dos modelos foi visto que existem diferenças significativas entre os modelos. Pela formalização matemática foi possível verificar que o método proposto possui efeito semelhante a regularização de Tikhonov. O modelo RN-ELM leva a função de separação para região de margem, sem a necessidade de cobrir exaustivamente todo o espaço de entrada, onde os parâmetros utilizados foram definidos pelas informações estruturais dos dados.

Palavras-chave: Classificador. Redes Neurais. Regularização. Treinamento com ruído.

Abstract

The learning capability of the artificial neural networks (ANN) depends the imposes constraints on solutions space which can be defined by the number of parameters of the model or by another constraints on it search space. To control the complexity of the neural network is used the decomposition the expectation of mean squared error into bias and variance terms of the model family. A technical used to control the tradeoff between bias and variance is the regularization that control the variance by a modification into error function by including a penalization term. The propose of this work is to define a classifier of large margin based on local resampling into training set in feature space. The thesis approach is based in the addition of noise during neural network training and on structural information of the data. Experiments were carried out to compare the proposed model called Regularization with Noise of Extreme Learning Machine (RN-ELM) against the standard Extreme Learning Machine (ELM) and Extreme Learning Machine with regularization (ELM-REG). The results showed that the methods RN-ELM and ELM-REG yield smoother solutions and decrease the weight norm. The Statistical test was applied on the mean accuracy of the models was observed that there are significative difference between the models. A mathematical formulation of the proposed method shows that the addition of synthetic samples has the same effect as the Tikhonov regularization. The RN-ELM approach leads the separation function to the margin region, without the need to exhaustively cover the whole input space and the parameters are fitted using the structural information of the samples.

Keywords: Classifier. Neural Networks. Regularization. Training with noise.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

CD	*Critical Difference*
CDD	*Critical Difference Diagram*
DNN	*Deep Neural Network*
ELM	*Extreme Learning Machine*
ELM-REG	*Extreme Learning Machine* com regularização
GAN	*Generative Adversarial Network*
GG	*Gabriel Graph*
MLP	*MultiLayer Perceptron*
MSE	*Mean Square Error*
PMCMR	*The Pairwise Multiple Comparison of Mean Ranks Package*
RBF	*Radial Basis Function*
RNA	Rede Neural Artificial
RN-ELM	*Regularization with Noise of Extreme Learning Machine*
SLFN	*Single hidden Layer Feedforward Neural network*
SSE	*Sum Square error*
SVM	*Support Vector Machine*

Lista de Símbolos

$\mathbf{x}$	Vetor
$\mathbf{w}$	Vetor de pesos
M	Matriz
$E[.]$	Esperança matemática ou valor esperado
$N(\mu, \sigma^2)$	Função de distribuição normal
$\tilde{w}$	Peso atualizado
w	Peso atual
$\|\cdot\|$	Norma Euclidiana
μ	Média
σ^2	Variância
J	Função objetivo
λ	Parâmetro de regularização
$\sum$	Somatório
$\omega(.)$	Termo de penalização
N_i	Número de amostra
$\tilde{J}_e$	Função de erro total
J_e	Função de erro padrão de saída
$f(.,.)$	Função de saída da rede neural
m	Número de dimensões
V	Conjunto de vértices do grafo
A	Conjunto de arestas do grafo
$\delta(\cdot)$	Distância Euclidiana entre dois pontos
$\mathbb{R}^m$	Vetor de números reais de m dimensões
L	Número de neurônios na camada oculta
y_i	Rótulo de saída
$g(.)$	Função de ativação

Sumário

Capítulo 1

Introdução

1.1 Contextualização

A definição da estrutura da rede neural artificial (RNA) para resolução de um problema não é uma tarefa fácil, pois depende da complexidade do problema, da dimensão dos dados de entrada, do conhecimento *a priori* sobre o problema e da representatividade dos dados. Muitos algoritmos de treinamento das redes neurais visam minimizar o risco empírico por meio da minimização do erro quadrático médio (*Mean Square Error*- MSE), com algoritmo de gradiente descendente (usado no *backpropagation*) não tendo como objetivo principal a maximização da margem entre as classes de um problema de classificação (Bishop, 1995a).

O processo de aprendizado pode ser visto como um problema de mapeamento não linear dos dados de entrada e seus rótulos e está relacionado com a capacidade do modelo de descrever os dados a partir de um conjunto de amostras da população. A RNA é avaliada pela sua capacidade de generalização que consiste em mapear corretamente os dados de entrada e saída de amostras da mesma população que nunca foram usadas no treinamento da rede neural (Haykin, 2009). Uma rede neural tem boa capacidade de generalização quando consegue identificar amostras que possuem pequenas diferenças do conjunto de treinamento da rede neural. As situações extremas do treinamento da rede neural são *overfitting* e *underfitting*, em que na primeira situação a rede neural memoriza as amostras de treinamento, inclusive os ruídos, o que deixa a rede incapaz de reconhecer amostras similares. Já a segunda situação no treinamento da rede neural ela não é capaz de realizar o mapeamento entre a entrada e rótulo (Duda *et al.*, 2000). No treinamento da rede neural não é necessário aprender uma exata representação do conjunto de treino, mas construir um modelo para entender como os dados são gerados. Portanto o estudo das redes neurais é motivado pela proposta de modelos que sejam capazes de buscar equilíbrio entre a complexidade da rede neural e a sua acurácia.

Otimizar a complexidade do modelo para obter uma boa generalização são requisitos primordiais no aprendizado de máquina. Para os modelos inflexíveis ou muito simples, tem-se um grande viés e uma pequena variância, o que significa que o modelo não está

aprendendo a verdadeira relação entre as variáveis e a predição de novas amostras. Outro problema é quando o viés é pequeno e a variância é grande, o modelo está muito ajustado ao conjunto de treino. Neste caso, o erro de generalização será grande, pois irá errar as variações entre o conjunto de dados. O viés e a variância são grandezas com restrições conflitantes e para se obter um bom modelo de generalização é necessário obter um compromisso entre os requisitos de minimização do viés e da variância, ver Figura 1.1 Para alcançar o equilíbrio, *trade-off*, entre viés e a variância é necessário controlar a complexidade do modelo que na rede neural pode ser feita pelo número de parâmetros da rede.

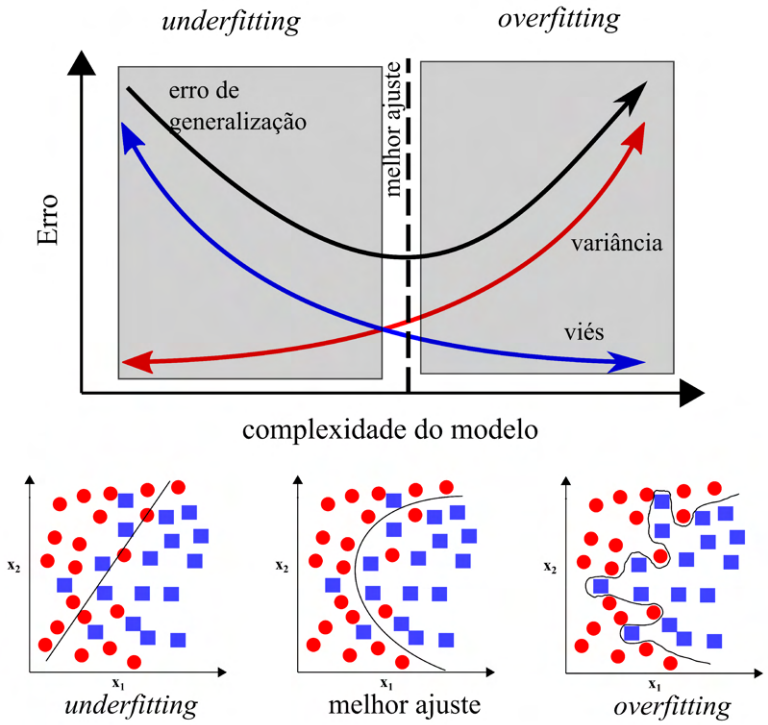


Figura 1.1: *Trade-off* entre viés e variância e seu efeito na superfície de separação nos casos de *underfitting*, melhor ajuste e *overfitting*.

Uma forma de ajustar a complexidade do modelo durante o treinamento é avaliar modelos com diferentes números de neurônios nas camadas ocultas. Alguns métodos são utilizados para determinar a arquitetura da rede neural como poda (Reed, 1993) (*prune*), método construtivo (PALMA Neto, 2004) e de regularização (Schlkopf *et al.*, 2018). O método de poda inicia a arquitetura da rede com grande número de neurônios e vai eliminando as conexões menos significativas até obter uma arquitetura adequada. Algumas desvantagens do método de poda são: não é possível determinar garantidamente uma arquitetura inicial que seja capaz de garantir a aproximação, desperdício computacional por eliminar neurônios que passaram por fases de processamento e não garante a seleção de uma arquitetura de menor dimensão. O método construtivo inicia o aprendizado com uma arquitetura mínima da rede e vai adicionando neurônios a medida que forem necessários

no treinamento, devido a construção dinâmica das camadas isto pode gerar redes pequenas. Tem como desvantagem a incorporação de ruído dos dados que adequam ao conjunto de treinamento. A Regularização é um método usado para controlar a complexidade do modelo e tende a evitar *overfitting* usando um modelo flexível com restrições dos valores que os parâmetros do modelo podem assumir, normalmente por meio da adição de um termo de penalização. A regularização pode ser obtida por diferentes formas pelos algoritmos de *weight decay* (Krogh & Hertz, 1992), *early stopping* (Hagiwara, 2002), adição de ruídos (Zur *et al.*, 2009) e *data augmentation* (Bouthillier *et al.*, 2016).

O método de adição de ruídos nos dados de entrada durante o treinamento de uma rede neural melhora a capacidade de generalização (Holmstrom & Koistinen, 1992; Matsuoka, 1992). No trabalho de Bishop (1995), foi visto que o treinamento com ruído é equivalente a regularização de Tikhonov. Desta forma, a minimização da função de erro regularizada é uma alternativa para o treinamento com ruído.

No trabalho de DeVries & Taylor (2017) transformações nos dados ocorrem no espaço de características onde a amostra gerada nem sempre tem a mesma classe da amostra original. A proposta *Smart Augmentation* (Lemley *et al.*, 2017) é usar GAN (*Generative Adversarial Network*) para aprender a gerar amostras com características combinadas que melhor descrevem um problema específico, e assim conseguem evitar o *overfitting* e aumentar o desempenho da rede neural.

Para criar modelos de redes neurais que melhor representem os dados de entrada, técnicas que exploram as informações estruturais dos dados são empregadas. O trabalho de Bennett & Bredensteiner (2000) descreve a interpretação geométrica do algoritmo de aprendizado de máquina *Support Vector Machine*(SVM) para dados linearmente separáveis, em que a margem máxima de separação entre dois conjuntos é equivalente a encontrar os pontos mais próximos sobre o feixe convexo de cada conjunto.

Para maximizar a margem de separação entre classes busca-se extrair informações geométricas dos dados de treinamento. No trabalho de Torres *et al.* (2015) é definido um grafo planar, grafo de Gabriel, para obter as informações geométricas dos dados de treinamento e identificar as amostras que estão mais próximas da margem de separação.

Assim, uma combinação a ser explorada neste trabalho é obter informações geométricas dos dados de treinamento para adicionar amostras sintéticas ruidosas na região de margem de separação das classes com o objetivo de obter o mesmo efeito de regularização de Tikhonov (Bishop, 1995). Para redes superdimensionadas o modelo deve ser capaz de descrever os dados de treinamento e suavizar a superfície de separação evitando o *overfitting* o que melhora a capacidade de generalização da RNA.

1.2 Problema de pesquisa

A definição de uma rede neural artificial (RNA) que seja adequada para tratar problemas de classificação e regressão tem sido um grande desafio e uma área muito estudada. O conjunto de amostras que trata o problema muitas vezes não é grande o suficiente para que a RNA aprenda o problema e possa generalizar para casos não conhecidos.

A qualidade dos modelos degradam em problemas de classificação com base de dados pequenas e desbalanceadas. As bases de dados pequenas afetam a capacidade de generalização enquanto as bases com classes desbalanceadas tornam enviesado o classificador para a classe dominante (Silva & Adeodato, 2011). Em problemas reais como casos de fraude, casos de doenças e regiões de simulações de larga escala as classes de interesse são as de menor ocorrência.

É de grande interesse a busca de métodos para RNA que sejam capazes de aprender com os dados de entrada e realizar previsões corretas de amostras não conhecidas dos modelos, ou seja, modelos com grande capacidade de generalização. A regularização tem sido estudada no contexto de redes neurais, normalmente usada para melhorar o desempenho de generalização quando o número de amostras é relativamente pequeno ou contaminado por ruído. Foi visto também que a adição de ruídos durante o treinamento da RNA pode obter o efeito de regularização (Bishop, 1995; Chandra & Singh, 2003; Rifai *et al.*, 2011).

Outro desafio é definir a complexidade da rede que seja capaz de relacionar um conjunto de características a uma saída esperada. Quanto mais complexa a RNA, maior a sua capacidade para resolver problemas. Essa rede complexa é definida pelo número de neurônios nas camadas ocultas e o grande número de parâmetros da rede que devem ser ajustados. RNAs complexas exigem mais tempo de treinamento e um conjunto de dados que sejam capazes de descrever corretamente o problema uma vez que a rede neural também aprende os ruídos que são amostras de dados que não representam as características do problema.

A capacidade da rede neural depende das restrições impostas ao espaço de soluções, podendo estas serem determinadas pelo número de parâmetros do modelo ou por outras formas de restrições de busca no espaço de soluções. No trabalho de Geman *et al.* (1992) pode ser visto que a variabilidade da família de modelos aproximadores influencia na qualidade da função de aproximação. A decomposição da esperança do erro quadrático, apresentada no trabalho de Geman resulta na Equação 1.1, que é conhecida por apresentar a decomposição do erro de aproximação $E[(f(\mathbf{x}, \mathbf{w}) - E[y|x])^2]$ em dois termos que representam o viés e a variância da família de modelos, onde $\mathbf{x}$ representa a amostra de entrada e y é a saída desejada e $E[.]$ é a esperança matemática. O objetivo da aproximação é aproximar o valor esperado de saída, $E[y|x]$ e, assim ambos os termos $(E[f(\mathbf{x}, \mathbf{w})] - E[y|x])^2$ (viés) e $E[(f(\mathbf{x}, \mathbf{w}) - E[f(\mathbf{x}, \mathbf{w})])^2]$ (variância) devem ser minimizados.

$$E[(f(\mathbf{x},\mathbf{w}) - E[y|x])^2] = \underbrace{(E[f(\mathbf{x},\mathbf{w})] - E[y|x])^2}_{\mu_{\{f(\mathbf{x},\mathbf{w})\}}} + \underbrace{E[(f(\mathbf{x},\mathbf{w}) - E[f(\mathbf{x},\mathbf{w})])^2]}_{\sigma^2_{\{f(\mathbf{x},\mathbf{w})\}}} \quad (1.1)$$

A fim de melhorar a capacidade de generalização da rede neural e ajustar a complexidade da rede de acordo com o problema de entrada, este trabalho propõe um método que utiliza as informações geométricas dos dados para adicionar amostras sintéticas ruidosas no conjunto de treinamento. Os vetores próximos a margem de separação das classes são identificados a partir da estrutura geométrica dos dados e serão utilizados como referência para definição das regiões de adição das amostras ruidosas no conjunto de treinamento. Desta forma, busca-se o efeito de regularização para que seja possível reduzir o erro de generalização e maximizar a margem de separação entre as classes nos problemas de classificação.

1.3 Hipótese

A hipótese deste trabalho é a de que a adição de amostras sintéticas ruidosas, no espaço de características, durante o treinamento de uma rede neural artificial possa maximizar a margem de separação entre as classes e melhorar a capacidade de generalização da rede neural. Os parâmetros serão definidos a partir das informações estruturais dos dados de treinamento, como a região de adição das amostras sintéticas, a dispersão das amostras sintéticas e a quantidade de amostras a ser adicionada.

1.4 Objetivos

1.4.1 Objetivos gerais

Propor um modelo de classificação que utilize as informações geométricas dos dados de entrada e seja capaz de maximizar a margem de separação entre as classes a partir da adição de amostras sintéticas ruidosas no conjunto de treinamento.

1.4.2 Objetivos específicos

1. Propor um modelo de classificação com adição de amostras sintéticas ruidosas.
2. Utilizar as informações geométricas dos dados de entrada para encontrar a margem de separação.

3. Definir a formalização matemática do modelo proposto que busca ter o mesmo efeito da regularização de Tikhonov.

4. Melhorar a capacidade de generalização da rede neural.

5. Maximizar a margem de separação entre as classes.

1.5 Organização do texto

O restante deste texto está organizado da seguinte forma: O Capítulo 2 apresenta a revisão bibliográfica, no capítulo 3 é apresentada a proposta do trabalho, no capítulo 4 os experimentos e resultados e no capítulo 5 a conclusão e os trabalhos futuros.

Capítulo 2

Revisão bibliográfica

Neste capítulo serão apresentados os principais métodos para melhorar a generalização no treinamento de uma rede neural artificial, as técnicas de regularização e a influência da adição de ruídos durante o treinamento de uma rede neural artificial.

2.1 Treinamento com ruído

Nesta seção serão abordados os principais estudos sobre a adição de ruído e suas aplicações. A adição de ruído durante o treinamento de uma rede neural artificial pode melhorar sua capacidade de generalização (Holmstrom & Koistinen, 1992; Matsuoka, 1992; Sum & Leung, 2021; Wang & Principe, 1999). Métodos de adição de ruídos podem evitar *overfitting* durante o treinamento de uma rede neural artificial (An, 1996; Zur *et al.*, 2009) e reduzir a complexidade da rede neural (Sum & Leung, 2021). Foi demonstrado por Bishop (1995) que o efeito de regularização de Tikhonov pode ser obtido com a adição dos ruídos durante o treinamento da rede neural.

O ruído injetado nos rótulos da base de dados permite desenvolver métodos de filtragem de ruído e algoritmos que sejam capazes de tratar ruído no rótulo (Garcia *et al.*, 2015, 2019; Tanaka *et al.*, 2018). No trabalho de An (1996) foi realizado um estudo sobre os efeitos da injeção de ruídos de três formas: ruído nos dados (2.1), ruído nos pesos (2.2) e ruído de *Langevin* (2.3).

1. Ruídos nos dados: $\mathbf{z}^t \rightarrow \mathbf{z}^t + \zeta$ (2.1)
2. Ruídos nos pesos: $\mathbf{w} \rightarrow \mathbf{w} + \xi$ (2.2)
3. Ruído Langevin: $\boldsymbol{\Delta}\mathbf{w} \rightarrow \boldsymbol{\Delta}\mathbf{w} + \xi$ (2.3)

onde ζ é um vetor de ruídos com a mesma dimensão de z^t, e ξ tem a mesma dimensão de $\mathbf{w}$. A perturbação dos dados de entrada pode ser por adição ou multiplicação, ou seja, a primeira quando um ruído gaussiano é adicionado aos atributos de entrada e a segunda é proporcional a magnitude dos atributos da entrada (Isaev & Dolenko, 2018).

Os experimentos foram realizados em problemas de classificação e regressão em que foram avaliados como os ruídos afetam a função de custo de aprendizado. Foi avaliado como a função de penalização induzida pelos ruídos afeta o desempenho de generalização no treinamento. Para injeção de ruídos nos dados foram utilizadas distribuições gaussianas $N(\mu, \sigma^2)$ e uma distribuição uniforme em um intervalo $[-a, a]$. A função de erro padrão possui dois termos de penalidade um idêntico ao termo de suavização da teoria de regularização e o segundo dependente do ajuste dos resíduos. O termo de penalização induzido pelo ruído na entrada fazem com que a rede neural seja menos sensível a variação dos dados de entrada. Desta forma o efeito de suavização melhora o desempenho de generalização. O ruído adicionado na saída não melhora a generalização, porque define uma constante na função de custo. Os ruídos adicionados no peso da rede alteram a função de custo de forma similar a penalização feita pelos ruídos na entrada. As restrições na rede neural induzida pelos pesos da rede tem efeito de generalização distintos, pois deixam a rede menos sensível a variação dos pesos em relação a variação dos dados de entrada. O treinamento com os ruídos de *Langevin* não tem efeito de regularização, mas é capaz de encontrar o mínimo local.

2.1.1 Treinamento com ruído em redes neurais profundas

A adição de ruídos em redes neurais profundas (DNN - *Deep neural network*) tem sido proposta para obter o efeito de regularização por *dropout*[1], melhorar a capacidade de generalização e tornar as redes mais robustas à presença de ruídos. Para o reconhecimento de fala, os modelos de DNN são utilizados devido a flexibilidade no aprendizado de padrões complexos (Yin *et al.*, 2015). Os dados da fala com adição de ruídos são usados no treinamento da DNN que aprende o padrão de ruído e compensa a interferência na fala. Desta forma, a perturbação gerada pelos ruídos pode melhorar a capacidade de generalização do modelo.

Um desafio para as DNN são aplicações de reconhecimento de imagem e *rank* de similaridade, devido aos diversos processamentos na imagem como compressão, mudanças de escalas e recortes. Para obter estabilidade no processamento da DNN, um método de treinamento baseado na adição de ruído gaussiano não correlacionado a nível de pixel na imagem de entrada é proposto por Zheng *et al.* (2016). No trabalho de Noh *et al.* (2017), para tratar aplicações de visão computacional, os ruídos são adicionados em unidades ocultas determinísticas para formar unidades estocásticas e explorar as formulações probabilísticas com o *Importance Weighted Stochastic Gradient Descent* que melhoram o desempenho de acurácia do modelo de DNN. Um classificador de margem larga é proposto por You *et al.* (2019) que para problemas de classificação visual nas rede neurais

[1]Técnica utilizada para remover aleatoriamente neurônios e suas conexões (entrada e saída) da rede neural durante o treinamento (Srivastava *et al.*, 2014).

convolucionais, um método com adição de ruídos no treinamento, faz com que aumente a perda de entropia cruzada e movimenta a superfície de separação para uma região mais distante das amostras de treinamento, aumentando assim a margem de separação.

A capacidade de reduzir o espaço de dimensionalidade tem demonstrado grande potencial do grafo *autoencoder*. Os dados de entrada são incorporados no grafo por uma matriz de fatorização que preserva a estrutura do grafo da matriz de entrada. A proposta de Wang *et al.* (2021b) é desenvolver uma estratégia de treinamento com adição de ruídos que utiliza o treinamento clássico do grafo de *autoencoder*. A abordagem consiste em remover aleatoriamente um conjunto de arestas e adicionar o mesmo número de arestas, sem alterar o número das mesmas.

2.1.2 Ruído nos dados de entrada

A possibilidade de melhorar a capacidade de generalização na rede neural com adição de ruídos nas amostras do conjunto de treinamento tem sido amplamente estudada. No trabalho de Holmstrom & Koistinen (1992) o algoritmo de treinamento utilizado é o *back-propagation* e a adição de ruído no conjunto de treinamento é interpretada como uma estimativa de *kernel* da densidade de probabilidade que descreve a distribuição dos vetores de treinamento. As características dos ruídos injetados dependem da definição da função *kernel* e do parâmetro de suavização para controlar a intensidade do ruído. Os experimentos realizados mostram a melhora na capacidade de generalização. Um problema crucial no algoritmo de *back-propagation* durante o treinamento é a capacidade de generalização. Essa dificuldade é justificada por selecionar um conjunto de treinamento limitado em relação a população que seja capaz de mapear entrada/saída de amostras desconhecidas do treinamento.

Assim no trabalho de Matsuoka (1992) é feito o desenvolvimento matemático que explica a ocorrência de suavização no mapeamento do espaço de entrada para o espaço de saída que melhora a capacidade de generalização no aprendizado usando *back-propagation*. No trabalho de Piotrowski & Napiorkowski (2013) a rede neural artificial é utilizada como uma ferramenta para hidrologia. A adição de ruído na entrada dos dados durante o treinamento é feita para evitar *overfitting*. A implementação em problemas reais é difícil e o desempenho do modelo depende de um grande número de detalhes técnicos que podem limitar sua aplicação prática. As amostras de treinamento onde o ruído deve ser adicionado são escolhidas por uma estimação da função de densidade do vetor de treinamento. Portanto, deve ser definida a forma da função de *kernel* e o parâmetro de suavização.

2.1.3 Ruído nos pesos da rede neural

A injeção de ruído nos pesos durante o treinamento é proposto com o objetivo de melhorar a generalização, a convergência e a tolerância a falhas da rede neural. Nos trabalhos de Ho *et al.* (2008) e Sum *et al.* (2012) é analisada a convergência do treinamento da *radial basis function* (RBF) com injeção de ruídos nos pesos e a função objetivo após a adição do ruído. Foi visto que com a injeção de ruído por multiplicação ou por adição a função a ser minimizada é a média dos erros quadráticos, como mostra a Equação 2.4 de ajuste dos pesos.

$$\tilde{w}_i(t) = \begin{cases} w_i(t) + \beta_i & \text{para injeção de ruído por adição,} \\ w_i(t) + \beta_i.w_i(t) & \text{para injeção de ruído por multiplicação.} \end{cases} \tag{2.4}$$

onde $\tilde{w}_i$ é o peso atualizado, w_i é o peso atual e β_i é uma variável gaussiana aleatória com média zero e variância dada por S_b de valor pequeno.

Portanto, a adição de ruído *online* nos pesos da rede RBF não melhora a tolerância a falha ou a generalização. Na rede *multilayer perceptron* (MLP), com o método de adição de ruído nos pesos, a função objetivo possui dois termos: o primeiro sendo a média dos erros quadráticos e o outro um termo regularizador da magnitude da saída da camada oculta. Um estudo dos efeitos no algoritmo de aprendizado causados pela adição de ruídos nos pesos foi feito por Sum *et al.* (2012), em que é combinada a injeção de ruído nos pesos com *weight decay* durante o treinamento. Foi demonstrado que para injeção de ruídos por adição ou multiplicação nos pesos da rede MLP, os algoritmos convergem com probabilidade 1, desde que o tamanho do passo atenda a certas condições: seja $J(\mathbf{w})$ a função objetivo do algoritmo de treinamento, sendo $\alpha > 0$ a constante de *weight decay* e $\mu(t)$ o tamanho do passo. Se $\mu(t) \to 0$, então tem probabilidade igual a 1, $E[||\mathbf{w}||_2^2]$ é limitada e $\exists \lim_{t\to\inf} ||\mathbf{w}(\mathbf{t})||_2$. A partir dessas duas propriedades mostrou-se que se $\mu(t) \to 0$, $\sum_t \mu(t) = \inf$ e $\sum_t \mu(t)^2 < \infty$ então com probabilidade igual a um esses algoritmos convergem. Além disso, $\mathbf{w}(t)$ converge com probabilidade igual a um para um ponto onde $\nabla_{\mathbf{w}} J(\mathbf{w}) = 0$.

2.1.4 Ruído no rótulo

Foi proposto por Wang & Principe (1999) a adição de ruído no valor da saída desejada (*target output*). O termo de ruído adicionado tem uma distribuição gaussiana com média zero e a variância independente dos sinais de entrada e saída desejada. Na abordagem de Wang & Principe, durante o treinamento, o ruído adicionado tem influência na função do erro total, mas não afeta o valor final dos pesos da rede neural. Durante o treinamento houve melhora na capacidade de busca pelo tamanho do passo do algoritmo de *back-propagation*. O método é simples e eficaz para pegar o processo de aprendizado em um mínimo local. Entretanto, não há garantia que uma solução ótima seja encontrada pois a

superfície de busca pode ter diferenças significativas mesmo que o ruído adicionado tenha restrição de média zero.

O tratamento de injeção de ruídos no rótulo aumenta a complexidade do modelo, aumenta a demanda de processamento e reduz a capacidade de predição do classificador para novas amostras. Desta forma, o pré-processamento da base de dados se faz importante para melhorar a qualidade dos dados e reduzir os efeitos prejudiciais no processo de aprendizado. No trabalho de Garcia *et al.* (2019) foram propostos modelos de injeção de ruídos no rótulo nas bases de dados de classificação. Estes modelos são capazes de produzir base de dados com ruídos mais realistas com a perturbação no rótulo de amostras críticas situadas próximas da superfície de decisão e também melhoram a avaliação da filtragem de ruídos.

No estudo de Garcia *et al.* (2015) foi avaliado como os ruídos afetam a complexidade nos problemas de classificação. A sensibilidade dos índices de complexidade dos dados foram monitorados em relação a diferentes níveis de ruídos. Para caracterizar a complexidade de um problema de classificação foram feitas medições geométricas, estatísticas e estruturais extraídas dos dados. Com a identificação das medidas mais sensíveis dos ruídos no rótulo foi possível desenvolver técnicas de pré-processamento e algoritmos mais tolerantes aos ruídos. Foi proposto um novo filtro de identificação de ruídos no rótulo com base nas medidas de complexidade.

O treinamento com ruído tem efeito equivalente a uma forma de regularização que adiciona um termo extra a função de erro. Entretanto, o termo de regularização que trata da segunda derivada da função de erro não é limitado inferiormente, o que o torna difícil de ser usado diretamente no algoritmo de aprendizado baseado na minimização do erro. Foi proposto por Bishop (1995) que o termo de regularização seja positivo e com a forma que envolva somente a primeira derivada no mapeamento da rede. Desta forma, com a injeção de ruídos nas amostras do conjunto de treinamento a função de somatório de erros quadráticos tem o termo de regularização pertencente a classe dos regularizadores de Tikhonov.

2.1.5 Conclusão

A injeção de ruídos nos dados de entrada durante o treinamento da rede neural melhora o desempenho de generalização da rede. Os ruídos adicionados nos dados e nos pesos da rede auxiliam para que os modelos sejam mais resilientes a variação dos dados de entrada, o que os torna menos sensíveis à pequenas variações dos dados. Os ruídos adicionados nos rótulos auxiliam no desenvolvimento de filtros e a desenvolver modelos tolerantes a falhas. Os filtros executam um pré-processamento na base de dados para tornar os dados mais confiáveis e, desta forma, é possível desenvolver modelos mais realistas.

As abordagens de adição de ruídos tratadas na literatura consistem em alterar os dados de entrada, os pesos e os rótulos. Na abordagem proposta neste trabalho é feita a adição de uma nova amostra sintética ruidosa ao conjunto de treinamento baseada em um subconjunto do conjunto de treinamento. Este subconjunto de amostras é definido de acordo com suas características, como os pontos situados na região da margem de separação.

2.2 Regularização

Regularização é uma técnica utilizada para controle do viés e variância (*bias and variance*) do modelo. O viés é a diferença entre a esperança de predição do modelo e o valor correto de saída do modelo, já a variância mede a variabilidade de predição do modelo para uma determinada base de dados (Bishop, 1995a). São normalmente utilizadas para evitar *overfitting*[1] no mapeamento da entrada para saída do modelo.

2.2.1 *Early stopping*

Uma forma simples de evitar *overfitting* é com a interrupção do processo de treinamento quando o erro no conjunto de validação começa a aumentar (Murphy, 2012). Durante o treinamento com o algoritmo *back-propagation* a complexidade da função de mapeamento tende a aumentar com a evolução das épocas. O *overfitting* pode ser identificado por meio da validação cruzada, em que as amostras do conjunto de treinamento são divididas em conjunto de estimação e conjunto de validação. O conjunto de estimação treina a rede e o conjunto de validação testa a rede para detectar quando o *overfitting* começa durante o treinamento supervisionado (Haykin, 2009; Prechelt, 2012). Na Figura 2.1 é apresentada a forma conceitual das curvas de aprendizado dos subconjuntos de estimação e de validação, e é indicado o ponto de parada no treinamento, chamado de *early stopping*

2.2.2 *Weight decay*

O *weight decay* é um método utilizado para melhorar a generalização da rede por meio da minimização da magnitude dos pesos da rede neural (Duda *et al.*, 2000). Grandes valores de pesos fazem com que o modelo tenha um ajuste irregular, mapeando os ruídos, causando *overfitting*. A abordagem do método consiste em iniciar uma rede com muitos pesos ou neurônios na camada oculta e realizar o decaimento dos pesos durante o aprendizado da rede neural. A regularização deste procedimento consiste em adicionar à função de custo,

[1] Modelo que aprende as amostras de treinamento (classificação perfeita), mas não é capaz de classificar corretamente novas amostras (Duda *et al.*, 2000).

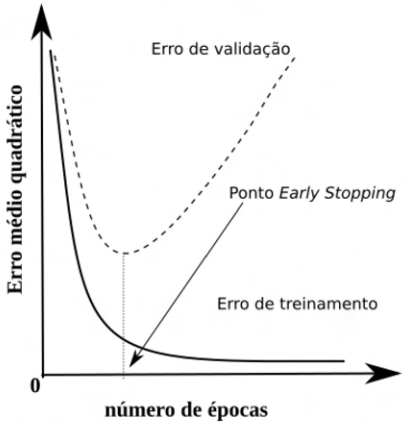


Figura 2.1: *Early-stopping* baseado no *cross-validation*

um termo de penalização da complexidade definido como a norma do vetor de pesos, $\mathbf{w}$, da rede (Krogh & Hertz, 1992), como mostra a Equação 2.5.

$$E(\mathbf{w}) = E_0(\mathbf{w}) + \lambda \sum_i w_i^2 \tag{2.5}$$

onde E é a função de custo, E_0 é uma função da medição do erro (ex. soma dos erros quadráticos), λ é o parâmetro de decaimento e $\mathbf{w}$ é o vetor com todos os parâmetros da rede. É comum o uso de validação cruzada (*cross-validation*) para selecionar o parâmetro λ (Murphy, 2012). Este processo força os pesos que tenham pouca influência no desempenho da rede a tomar valores próximos de zero, enquanto outros de maior significância mantenham valores maiores (Haykin, 2009).

2.2.3 Regularização de Tikhonov

A regularização de Tikhonov é um método utilizado para resolver sistemas lineares mal postos e problema de mínimos quadrados. Devido à importância de se definir o parâmetro de regularização vários métodos são definidos com esse propósito (Golub & Von Matt, 1997). Um exemplo é utilizar o critério da curva em L para seleção de parâmetros no método de regularização de Tikhonov (Braga, 2001; Calvetti *et al.*, 2000).

Os problemas inversos tem como objetivo descobrir características desconhecidas de um objeto a partir da observação de uma resposta deste objeto. Uma características dos problemas inversos é que também são mal posto e sua solução é muito sensível a variações da entrada (Borges & Bazan, 2009).

Foi demonstrado por Golub *et al.* (1999a) a discretização de problemas inverso em sistemas de equações com coeficientes da matriz mal condicionados. O método de regularização de Tikhonov será aplicado a uma classe de métodos que produzem uma aproximação da solução para um sistema linear de equações do tipo $\mathbf{Xw} \approx \mathbf{y}$ e substitui a Equação 2.6 pela Equação 2.7 (Bazan, 2009).

$$\mathbf{w} = \arg \min_{\mathbf{w} \in \mathbb{R}^m} ||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 \tag{2.6}$$

por

$$\mathbf{w} = \arg \min_{\mathbf{w} \in \mathbb{R}^m} \{||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2 + \lambda^2 ||L(\mathbf{w} - \mathbf{w}_0)||_2^2\} \tag{2.7}$$

em que λ é o parâmetro de regularização, $||\mathbf{y} - \mathbf{X}\mathbf{w}||_2^2$ é o erro de aproximação, $||L(\mathbf{w} - \mathbf{w}_0)||_2^2$ é o termo de penalização e $\mathbf{w}_0$ é a aproximação inicial da solução quando disponível, mas se não tiver informação inicial tem-se $\mathbf{w}_0 = 0$. A matriz L é normalmente a matriz identidade $L = \mathbf{I}$ ou uma aproximação discreta de algum operador diferencial definido, por exemplo, a 1^a ou 2^a derivada. O desafio é escolher λ tal que $\mathbf{w}$ se aproxime da solução ótima representada por $\mathbf{w}^*$. A Equação 2.7 é resolvida como um problema de mínimos quadrados e representada na equação normal, Equação 2.8.

$$(\mathbf{X}^T\mathbf{X} + \lambda^2 L^T L)\mathbf{w} = \mathbf{X}^T\mathbf{y} + \lambda^2 L^T L\mathbf{w}_0 \tag{2.8}$$

esta é a equação normal regularizada, caso seja considerado $L = \mathbf{I}$ então o problema está na forma padrão, caso contrário está na forma geral. Considerando $\mathbf{w}_0 = 0$ a Equação 2.8 pode ser reescrita como

$$\mathbf{w} = (\mathbf{X}^T\mathbf{X} + \lambda^2 L^T L)^{-1}\mathbf{X}^T\mathbf{y} \tag{2.9}$$

A técnica de regularização de Tikhonov tem como objetivo suavizar o mapeamento da rede pela adição de um termo de penalidade, $\lambda\Omega(.)$, como mostra a função de erro 2.11 (Bishop, 1995b; de Campos Velho, 2001; Poggio & Girosi, 1990).

$$J_e = \sum_{i=1}^{N_1} (\mathbf{y}_i - f(\mathbf{x}_i, \mathbf{w}))^2 \tag{2.10}$$

onde N_1 é o número de amostras no conjunto de treinamento.

O termo de penalização penaliza o mapeamento que não é suavizado. E o efeito da regularização na rede neural permite controlar o sobreajuste, *overfitting*, que ocorre quando a capacidade do modelo excede a complexidade do problema. A expressão de regularização pode ser descrita na Equação 2.11.

$$\tilde{J}_e = J_e + \lambda\Omega(f(\mathbf{x}_i, \mathbf{w})) \tag{2.11}$$

onde $\tilde{J}_e$ é a função do erro total, J_e é o erro padrão de saída, λ é o parâmetro de regularização que controla o compromisso entre a suavidade da solução e a aproximação dos dados e o Ω é o termo de regularização que geralmente é expresso em termos da função da rede $f(\mathbf{x}_i, \mathbf{w})$ (Bishop, 1995).

2.2.4 Conclusão

A regularização é uma importante técnica para melhorar o desempenho da generalização. Consiste em utilizar mecanismos de controle dos pesos da rede para ajustar o modelo as características dos dados de entrada durante o aprendizado da rede. Para o método *early stopping* nem sempre é uma tarefa fácil decidir quando exatamente parar o treinamento (Piotrowski & Napiorkowski, 2013). O treinamento pode ser interrompido imediatamente quando o erro de validação começa a aumentar ou quando for 20% maior que o valor mínimo do erro de validação. O algoritmo de *weight decay* propõe controlar a complexidade da rede com a penalização dos pesos, fazendo com que elementos de menor importância tenham valores próximo a zero e os mais significantes tenham maiores valores.

No estudo realizado por Bishop (1995) foi visto que o efeito de regularizaçao pode ser obtido com a injeção de ruídos nos dados durante o treinamento. Nossa proposta consiste em adicionar amostras sintéticas ruidosas nas proximidades da superfície de separação para obter um modelo que seja capaz de maximizar a margem de separação. Diferente da abordagem usada por Bishop (1995) que injeta ruídos nos dados de entrada, neste trabalho foram adicionadas novas amostras em regiões, no espaço de características, a fim de obter o mesmo efeito da regularização de Tikhonov.

2.3 Geometria computacional

2.3.1 Grafo de Gabriel

O Grafo de Gabriel (GG) (Gabriel, K. Ruben and Sokal, 1969) é um grafo planar onde os vértices são as amostras e as arestas definidas pela distância Euclidiana entre dois vértices de tal forma que nenhum outro vértice esteja mais próximo dos vértices conectados pela aresta. A proposta do GG é encontrar regiões de uma área de acordo com as características dos indivíduos presentes em diferentes localidades desta área. O GG é construído com base nas informações geométricas do conjunto de amostras de entrada $\mathbf{x} \in \mathbb{R}^m$ conexo que pode ser definido pelo conjunto de vértices $\mathbf{V} = \{\mathbf{x}_i\}_{i=1}^{N}$ e o conjunto de arestas formadas por $\mathbf{A} = \{(\mathbf{x}_i, \mathbf{x}_j) | i \neq j\}$ satisfazendo a seguinte inequação 2.12

$$\delta(\mathbf{x}_i, \mathbf{x}_j)^2 \leq [\delta(\mathbf{x}_i, \mathbf{x}_k)^2 + \delta(\mathbf{x}_j, \mathbf{x}_k)^2], \forall \mathbf{x}_k \in \mathbf{V} \text{ e } i \neq j \neq k \tag{2.12}$$

onde $\delta(.)$ é a distância euclidiana entre vetores.

A representação gráfica de uma aresta $(\mathbf{x}_i, \mathbf{x}_j)$, definida pela inequação 2.12 é mostrada na Figura 2.2. Já a Figura 2.3 apresenta uma aresta $(\mathbf{x}_i, \mathbf{x}_j)$ que não pertence ao GG, porque o vértice $\mathbf{x}_k$ está inserido dentro da circunferência entre $\mathbf{x}_i$ e $\mathbf{x}_j$. A Figura 2.5

apresenta o Grafo de Gabriel produzido a partir do conjunto de dados de duas classes da Figura 2.4.

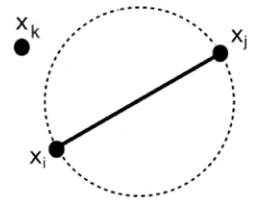


Figura 2.2: Aresta pertencente ao GG.

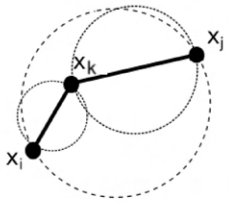


Figura 2.3: Aresta não pertencente ao GG.

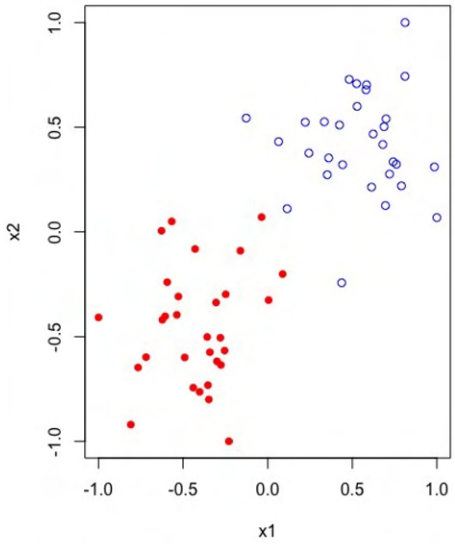


Figura 2.4: Base de dado sintética

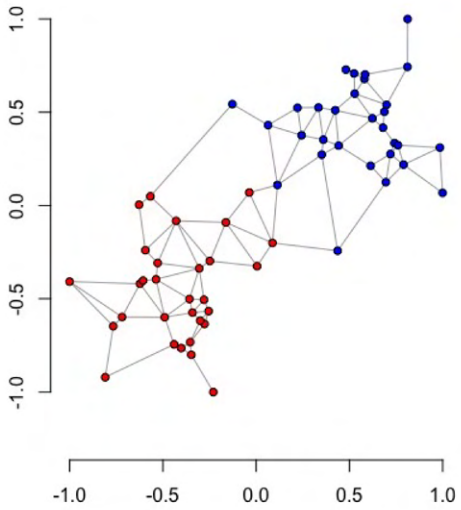


Figura 2.5: Grafo de Gabriel da base de dados sintética

A estrutura do GG utilizada neste trabalho é baseada na abordagem feita por Torres *et al.* (2015), que utiliza um método para eliminação de sobreposição entre as classes no GG. A Figura 2.6 apresenta um GG que foi produzido a partir de uma base de dados com sobreposição. Após aplicar o método de eliminação de sobreposição temos o GG apresentado na Figura 2.7. A sobreposição de amostras entre as classes representam ruídos das amostras, após remover os ruídos é possível definir os vértices de borda da região de separação das classes.

2.3.2 Conclusão

Um desafio para melhorar o desempenho de generalização do modelo de rede neural é extrair informações dos dados de entrada. O modelo tem que ser capaz de realizar o mapeamento eficiente dos dados de entrada/saída. O Grafo de Gabriel tem sido um método eficiente para descrever a relação dos dados. Por meio das informações do grafo é possível determinar regiões importantes como a margem de separação e, também é

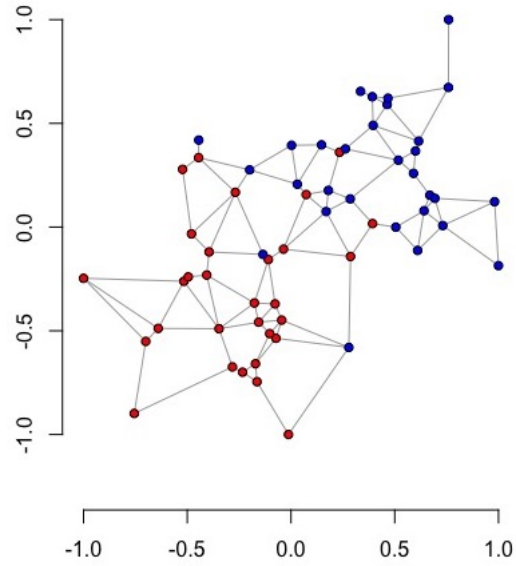


Figura 2.6: Grafo de Gabriel com sobreposição.

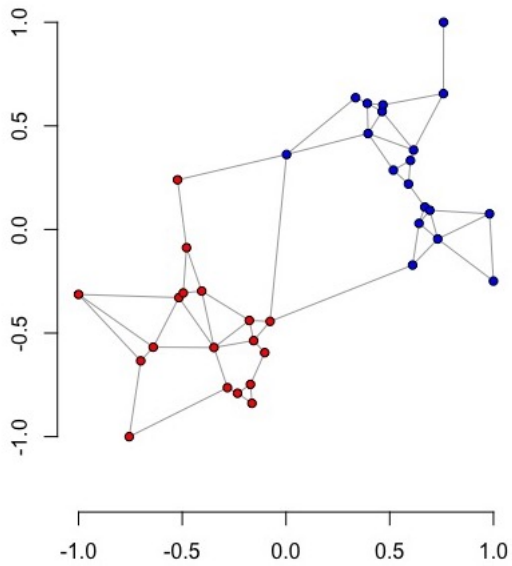


Figura 2.7: Grafo de Gabriel com eliminação da sobreposição.

possível remover vetores de sobreposição entre os dados para encontrar superfícies mais simples de separação.

2.4 Máquina de Aprendizado Extremo

A máquina de aprendizado extremo (ELM *Extreme Learning Machine*) foi proposta por Huang *et al.* (2004, 2006) e corresponde a um algoritmo de aprendizado de rede neural *feedforward* com uma única camada oculta (*Single hidden Layer Feedforward Neural network* - SLFN). É um método de fácil implementação em que os pesos e *bias* da camada oculta são determinados aleatoriamente e não são alterados durante o treinamento. Já os pesos da camada de saída são determinados analiticamente pela matriz inversa generalizada.

A rede ELM, Figura 2.8, tem sido estudada em vários trabalhos (Araujo *et al.*, 2019; Deng *et al.*, 2009; Ding *et al.*, 2014; Inaba *et al.*, 2017; Silvestre *et al.*, 2015) por causa de sua boa capacidade de generalização, treinamento rápido além de ser de fácil implementação. Com funções de ativação continuas e não lineares fazem com que a ELM tenha

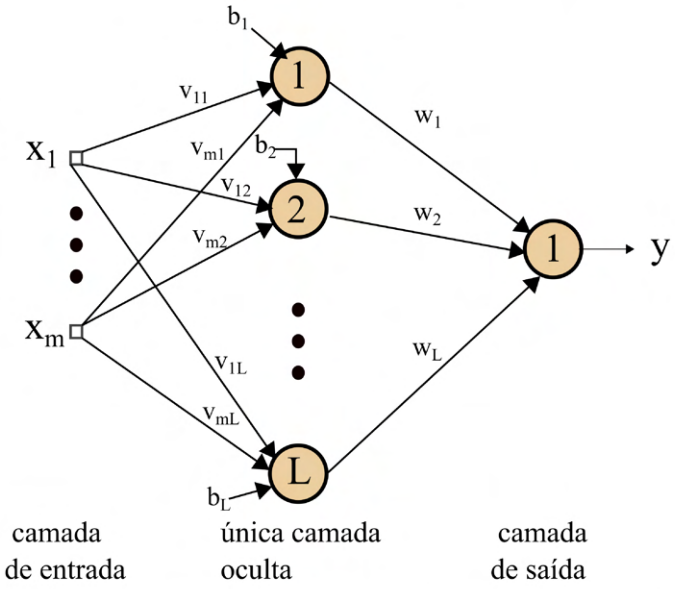


Figura 2.8: *Extreme Learning Machine* (ELM) com uma única saída.

a capacidade de aproximação universal, se o número de neurônios tende a infinito. Normalmente no treinamento das redes neurais os parâmetros da camada oculta são ajustados iterativamente. Já na ELM somente os pesos de saída são ajustados pela função Inversa de Moore-Penrose generalizada resolvendo um problema linear, Equação 2.14. A única informação para se definir na ELM é o número de neurônios da camada oculta. Os pesos de entrada e *bias* não são alterados durante o treinamento da rede e somente os pesos da camada de saída são ajustados (Wang *et al.*, 2021a).

Dado um conjunto de N amostras distintas $(\mathbf{x}_i, y_i)$ onde $\mathbf{x}_i = [x_{i1}, x_{i2}, \ldots, x_{im}]^T \in \mathbb{R}^m$ e $y_i \in \mathbb{R}$ para $i = \{1, \ldots, N\}$, a saída da rede neural é dada pela Equação 2.13

$$\hat{\mathbf{y}}_i = \sum_{j=1}^{L} w_j g_j(\mathbf{x}_i) = \sum_{j=1}^{L} w_j g(\mathbf{v}_j \mathbf{x}_i + b_j) \quad i = 1, \ldots, N \tag{2.13}$$

onde L é o número de neurônios na camada oculta, $g(.)$ é a função de ativação, $\mathbf{v}_j = [v_{j1}, v_{j2}, \ldots, v_{jm}]^T$ são os pesos que conectam a entrada na camada oculta e w_j são os pesos que conectam a camada oculta a camada de saída. Finalmente, b_j é o termo de *bias* do j-th neurônio da camada oculta. Para uma SLFN com L neurônios na camada oculta, que é capaz de aproximar para uma função de N amostras, então existe $\mathbf{w}, \mathbf{v}_j$ e b_j de tal modo que a Equação 2.14 é satisfeita.

$$\mathbf{Hw} = \mathbf{y} \tag{2.14}$$

onde

$$\mathbf{H} = \begin{bmatrix} g(\mathbf{v}_1\mathbf{x}_1 + b_1) & \ldots & g(\mathbf{v}_L\mathbf{x}_1 + b_L) \\ \ldots & \ldots & \ldots \\ g(\mathbf{v}_1\mathbf{x}_N + b_1) & \ldots & g(\mathbf{v}_L\mathbf{x}_N + b_L) \end{bmatrix}_{N\times L} \tag{2.15}$$

$$\mathbf{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_L \end{bmatrix}_{L\times 1} \qquad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_N \end{bmatrix}_{N\times 1} \tag{2.16}$$

A solução que minimiza a norma pela equação dos mínimos quadrados do sistema linear apresentado na Equação 2.14 é $\mathbf{w} = \mathbf{H}^{\dagger}\mathbf{y}$, onde $\mathbf{H}^{\dagger}$ é a matriz inversa generalizada de Moore Penrose da matriz $\mathbf{H}$ (Huang *et al.*, 2006). Quando a projeção ortogonal $\mathbf{H}^T\mathbf{H}$ é não singular, podemos escrever a matriz pseudo-inversa como $\mathbf{H}^{\dagger} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ (Barata & Hussein, 2012; Ferrari & Stengel, 2005). Desta forma, a matriz de melhor aproximação da solução pode ser escrita como

$$\mathbf{w} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y} \tag{2.17}$$

Assim, os pesos de saída $\mathbf{w}$ são calculados em uma única etapa o que evita o procedimento de um treinamento longo em que os parâmetros da rede neural são ajustados iterativamente até que as restrições de parada sejam atendidas (Zheng *et al.*, 2013).

2.4.1 ELM com regularização

O método da ELM utiliza o princípio de minimização do risco empírico, erro de treinamento, o que tende a obter um modelo com *overfitting* (Vapnik, 1995). Para tratar essa dificuldade do ELM é proposto por Deng *et al.* (2009) um modelo ELM regularizado, que se fundamenta na minimização do risco estrutural.

De acordo com a teoria do aprendizado estatístico, um modelo com boa capacidade de generalização deve controlar o *trade-off* entre o risco empírico e o risco estrutural

do modelo. O risco empírico é dado pela soma dos erros quadráticos ($||\boldsymbol{\varepsilon}||^2$) e o risco estrutural é dado pela norma dos pesos ($||\mathbf{w}||^2$) (Vapnik, 1995).

A ELM regularizada utiliza penalização L_2 ou seja, regularização de Tikhonov, para deixar o modelo mais robusto em relação a interferência de *outliers*. A variável de erro $||\boldsymbol{\varepsilon}||^2$ é ponderada pelo fator s_j e será estendida para $||D\boldsymbol{\varepsilon}||^2$. O termo de regularização γ é adicionado para realizar o ajuste do risco empírico e risco estrutural. A expressão de regularização pode ser definida por

$$\mathbf{w} = \left(\frac{\mathbf{I}}{\gamma} + \mathbf{H}^T D^2 \mathbf{H}\right)^{\dagger} \mathbf{H}^T D^2 \mathbf{y} \tag{2.18}$$

onde D é uma matriz diagonal com os pesos $s_1, s_2, \ldots, s_j$ onde $L << N$. Para a matriz $D = \mathbf{I}$ matriz identidade, temos a ELM regularizada não ponderada (*unweighted regularized ELM*):

$$\mathbf{w} = \left(\frac{\mathbf{I}}{\gamma} + \mathbf{H}^T \mathbf{H}\right)^{\dagger} \mathbf{H}^T \mathbf{y} \tag{2.19}$$

A ELM regularizada proposta por Huang *et al.* (2012), adiciona um termo de regularização, C no cálculo da pseudo-inversa. Nesta abordagem é analisada a relação do número de amostras de treinamento (N) e o número de neurônios na camada oculta (L). Obtém-se o cálculo dos pesos de saída dado pelo seguinte sistema:

$$\mathbf{w} = \begin{cases} \mathbf{H}^T \left(\dfrac{\mathbf{I}}{C} + \mathbf{H}\mathbf{H}^T\right)^{-1} \mathbf{y}, & \text{se } L > N \\ \left(\dfrac{\mathbf{I}}{C} + \mathbf{H}^T\mathbf{H}\right)^{-1} \mathbf{H}^T \mathbf{y}, & \text{se } L << N \end{cases} \tag{2.20}$$

onde o termo de regularização C é definido normalmente com validação cruzada. A Equação 2.20 leva em conta o custo computacional e sua eficiência de acordo com a aplicação. Quando o conjunto de treinamento for muito grande ($L << N$) o custo computacional por ser reduzido.

2.4.2 Conclusão

As redes neurais ELM tem sido amplamente estudadas e aplicadas em grande variedades de problemas. Por ser tratar de uma rede neural de treinamento rápido, a boa capacidade de convergência e a fácil implementação justificam o interesse de aplicação.

A simplicidade do seu treinamento, em que somente os pesos de saída são ajustados as tornam atrativas para implementação. A abordada deste trabalho consiste na reamostragem no espaço de característica da rede neural maximizando a margem de separação e a capacidade de generalização. A suavização da superfície de separação é obtida pelo efeito da regularização de Tikhonov. O modelo ELM regularizado proposto por Huang

et al. (2012) foi implementado para comparar o desempenho do modelo proposto. O ELM regularizado foi utilizado como referência para demonstrar que o modelo proposto neste trabalho tenha o efeito de regularização.

2.5 Lei dos Grandes números

Dado um conjunto de amostras que são independentes e identicamente distribuídas (iid) como a sequência $\{X_1, X_2, X_3, \ldots, X_n\}$, que segue uma distribuição que existe interesse em gerar novas variáveis aleatórias, $Y = h(X_1, X_2, X_3, \ldots, X_n)$, a partir de uma função h. A variável Y é estimada a partir de amostras que seguem uma dada distribuição. Desta forma, o conjunto de Y será usado para estimar a característica da distribuição de X_i, onde a média amostral $\overline{X}$ será uma estimativa para a média da distribuição de X_i. Se os estimadores forem bem definidos e um grande conjunto de amostras aleatórias forem geradas, ocorre convergência para a média do conjunto amostrado (Evans & Rosenthal, 2004).

2.5.1 Lei fraca dos grandes números

A lei fraca dos grandes números é uma importante aplicação de convergência em probabilidade. Suponha a sequência de variáveis aleatórias independentes $X_1, X_2, \ldots$ onde cada uma possui o mesmo valor esperado de μ e variância σ^2. Então, a média amostral, $\overline{X}_n$ converge em probabilidade para μ e expressa pela Equação 2.21 (Cao & Qiao, 2008; Evans & Rosenthal, 2004) .

$$\overline{X}_n = \frac{1}{n}(X_1 + X_2 + \ldots + X_n) \tag{2.21}$$

para todo $\varepsilon > 0$, não importa o quão pequeno ele seja.

$$\lim_{n\to\infty} P(|X_n - \mu| \geq \varepsilon) = 0 \tag{2.22}$$

Assim, de acordo com a lei fraca dos grandes números, para n muito grande o valor médio $\overline{X}_n$ converge em probabilidade para μ.

2.5.2 Lei forte dos grandes números

A lei forte dos grandes números conclui convergência com probabilidade 1 ao invés de convergir em probabilidade como na lei fraca dos grandes números. Considere a sequência de variáveis aleatórias independentes e igualmente distribuídas $X_1, X_2, \ldots, X_n$, onde cada uma tem média finita μ. Assim temos,

$$P\left(\lim_{n\to\infty} X_n = \mu\right) = 1 \tag{2.23}$$

o que significa que a média amostral converge com probabilidade 1 para a média μ.

2.5.3 Conclusão

As amostras sintéticas geradas em cada hiperesfera tem o objetivo de suavizar a superfície de separação reproduzindo o efeito de regularização. Com base na lei fraca dos grandes números a geração de um grande número de amostras aleatórias independentes e identicamente distribuídas faz com que a média amostral se aproxime da média populacional (Kruglov, 2011). No entanto, para reduzir o número de amostras geradas em cada hiperesfera foi proposto a simetria das amostras para aproximação da média populacional das mesmas.

Capítulo 3

Proposta

Neste capítulo será apresentado o método de regularização proposto com a adição de amostras sintéticas *Regularization with Noise of Extreme Learning Machine* (RN-ELM). O modelo proposto é baseado na estrutura geométrica dos dados de entrada que permite explorar a margem de separação entre as classes para o problema de separação. Será apresentada a formulação matemática que mostra que o modelo proposto se assemelha a regularização de Tikhonov.

3.1 Contextualização

A capacidade de uma rede neural artificial depende das restrições impostas ao seu espaço de soluções que podem ser determinadas pelo número de parâmetros do modelo ou por outras formas de restrições à busca neste espaço. Assim, um grande desafio no treinamento de uma rede neural é definir a complexidade ótima do modelo que seja capaz de classificar corretamente os dados desconhecidos do mesmo. Desta forma, várias propostas de algoritmos de aprendizado tem sido desenvolvidas para que redes neurais artificiais sejam capazes de se adaptarem aos problemas e melhorar a capacidade de generalização (Ludermir *et al.*, 2006; Zhang & Zhou, 2006).

Uma forma de controlar a complexidade da rede neural por meio da decomposição da esperança do erro quadrático que mostra a decomposição do erro de aproximação em dois termos, o viés e a variância da família de modelos (Geman *et al.*, 1992; Geurts, 2010). Uma técnica utilizada para controlar o *trade-off* entre o viés e a variância é a regularização que controla a variância pela modificação da função de erro com adição de um termo de penalização (Bishop, 1995c; Hagiwara, 2002; Poggio & Girosi, 1990).

3.1.1 Proposta do modelo RN-ELM

A proposta deste trabalho é um método de classificação baseado na reamostragem local, na margem de separação, que seja capaz de restringir o espaço de soluções da rede neural

e reduzir a complexidade do modelo. Para melhorar a capacidade de generalização do modelo busca-se obter o mesmo efeito da regularização de Tikhonov (Bishop, 1995) com a adição de amostras sintéticas no conjunto de treinamento. A região de adição das amostras sintéticas é baseada nos vetores de borda definidos por Torres *et al.* (2020), obtidos a partir das informações estruturais do conjunto de treinamento. As amostras sintéticas têm o objetivo de reforçar a borda da separação entre as classes. Nesta proposta não é necessário inserir ruídos em cada uma das amostras do conjunto de entrada durante o treinamento. Os vetores de borda são utilizados como um referencial para gerar as amostras sintéticas, as quais serão adicionadas ao conjunto de treinamento.

A Figura 3.1 mostra a adição das amostras sintéticas ruidosas dentro da hiperesfera. As amostras sintéticas foram geradas por uma função de distribuição normal, as amostras simétricas são referências espelhadas das amostras sintéticas em relação ao centro da hiperesfera (r). As principais referências geométricas foram destacadas como D a distância euclidiana entre os vértices de borda, r o centro da hiperesfera e ϵ são ruídos.

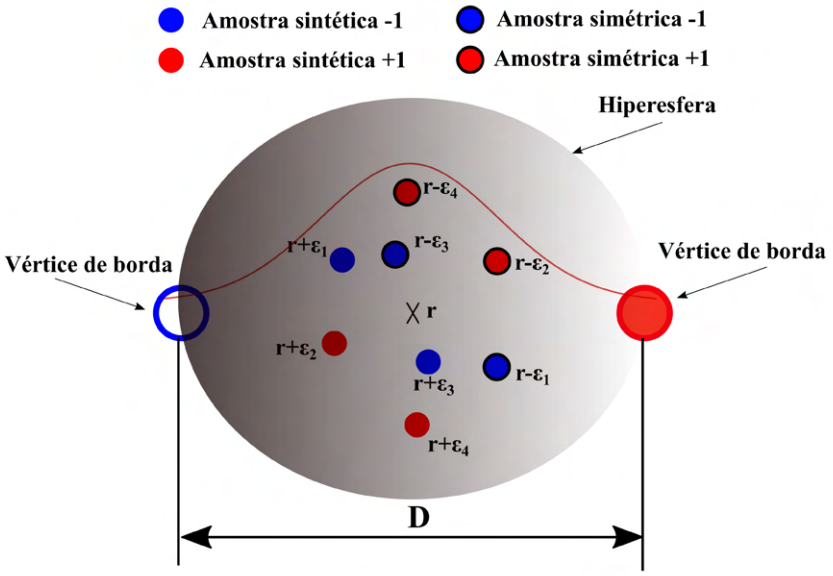


Figura 3.1: Adição de amostras sintéticas ruidosas na hiperesfera formada pela referência dos vértices de borda.

O Algoritmo 1 descreve o treinamento da rede ELM com o modelo RN-ELM, as linhas **9** e **10** do algoritmo representam o ajuste dos pesos da rede.

A informação estrutural dos dados de entrada foram extraídas de um grafo planar, o Grafo de Gabriel (GG), gerado a partir do conjunto de treinamento. Cada padrão do conjunto de treinamento será um vértice do grafo. Assim, dado um conjunto de dados $\mathbf{D} = \{\mathbf{x}_i, y_i\}_{i=1}^{N}$ com $\mathbf{x}_i \in \mathbb{R}^m$, $y_i \in \{-1, +1\}$, os vértices do GG são $\mathbf{V}_{GG} = \{\mathbf{x}_i \in \mathbf{D} \,|\, i = 1, \ldots, N\}$ e as arestas formadas por $\mathbf{E}_{GG} = \{(\mathbf{x}_i, \mathbf{x}_j) \,|\, \delta(\mathbf{x}_i, \mathbf{x}_j) \leq [\delta(\mathbf{x}_i, \mathbf{x}_k) + \delta(\mathbf{x}_j, \mathbf{x}_k)]$, onde $\mathbf{x}_i, \mathbf{x}_j, \mathbf{x}_k \in \mathbf{V}_{GG}$ com $i \neq j \neq k\}$ e $\delta(.)$ é o operador da distância euclidiana. O conjunto de arestas de borda ($\mathbf{E}_{BR}$) é definido por $\mathbf{E}_{BR} = \{(\mathbf{x}_i, \mathbf{x}_j) \,|\, x_i, x_j \in \mathbf{V}_{GG}$ e $y_i \neq y_j\}$, onde $\mathbf{E}_{BR} \subset \mathbf{E}_{GG}$.

Algoritmo 1: RN-ELM Treinamento com RN-ELM

Input: $\mathbf{X}_{treino}, \mathbf{y}_{treino}$: Conjunto de treinamento , $\mathbf{V}$: Pesos da camada oculta
Output: $\mathbf{w}_{saida}$: Pesos de saída da ELM

$acc_{max} \leftarrow 0$; // Maior acurácia
// 10-*folds* validação cruzada
for $k \leftarrow 1$ ***to*** 10 **do**
  $\mathbf{X}_{val} \leftarrow \mathbf{X}_{treino}[-k], \mathbf{y}_{val} \leftarrow \mathbf{y}_{treino}[-k]$
  $\mathbf{X}_{val\_teste} \leftarrow \mathbf{X}_{treino}[k], \mathbf{y}_{val\_teste} \leftarrow \mathbf{y}_{treino}[k]$
  $\mathbf{H} \leftarrow \phi(\mathbf{X}_{val}.\mathbf{V})$ // Espaço de características
  $vertices \leftarrow$ GG$(\mathbf{H}, \mathbf{y}_{val})$ // Função Grafo de Gabriel (GG)
  // Ruídos por classe dentro da hiperesfera
  for $i \leftarrow 1$ ***to*** 10 **do**
    $(\mathbf{R}, \mathbf{E}) \leftarrow$ GerarRuidos$(vertices)$
    $\Lambda = \mathbf{R^T R} + \mathbf{E^T E} + \mathbf{R^T E} + \mathbf{E^T R}$
    $\mathbf{w} \leftarrow (\mathbf{H^T H} + \mathbf{\Lambda})^{-1}(\mathbf{H^T y}_{val})$
    $\hat{\mathbf{y}} \leftarrow \phi(\mathbf{X}_{val\_teste}\mathbf{V})\mathbf{w}$
    $acc \leftarrow$ Media$(\hat{\mathbf{y}} = \mathbf{y}_{val_teste})$
    if $acc > acc_{max}$ **then**
      $\mathbf{w}_{saida} \leftarrow \mathbf{w}$
return $\mathbf{w}_{saida}$

O método de filtragem de sobreposição no GG foi proposto por Torres (2016) e baseia-se no grau do vértice. A medida de qualidade de cada vértice é determinada pela razão entre o número de arestas que conecta vértices de mesma classe pelo número total de arestas do vértice. Um limiar de remoção do vértice no grafo é definido como a razão entre a soma da qualidade do vértice de um grupo pelo número de vértices dentro do grupo, em que o grupo é formado por vértices de mesma classe. Assim, os vértices que tiverem índice de qualidade menor que o limiar definido serão removidos. Após a remoção dos vértices sobrepostos é definido um novo grafo, em que serão obtidos os vértices de borda para adição das amostras sintéticas no conjunto de treinamento.

3.1.2 Formulação matemática

Para avaliar o ajuste do modelo ao conjunto de treinamento foi utilizada a expressão geral da soma dos erros quadráticos (*Sum Square Error* - SSE) descrita na Equação 3.1.

$$J_e = \sum_{i=1}^{N_1}(y_i - f(\mathbf{h}_i, \mathbf{w}))^2 \tag{3.1}$$

onde N_1 é o número de amostras no conjunto de treinamento.

Para exemplificar a adição de amostras sintéticas na região de borda o conjunto de treinamento foi definido por meio da base de dados *two moons*, mostrada na Figura 3.2. O GG é gerado a partir do conjunto de treino onde cada vértice representa uma amostra

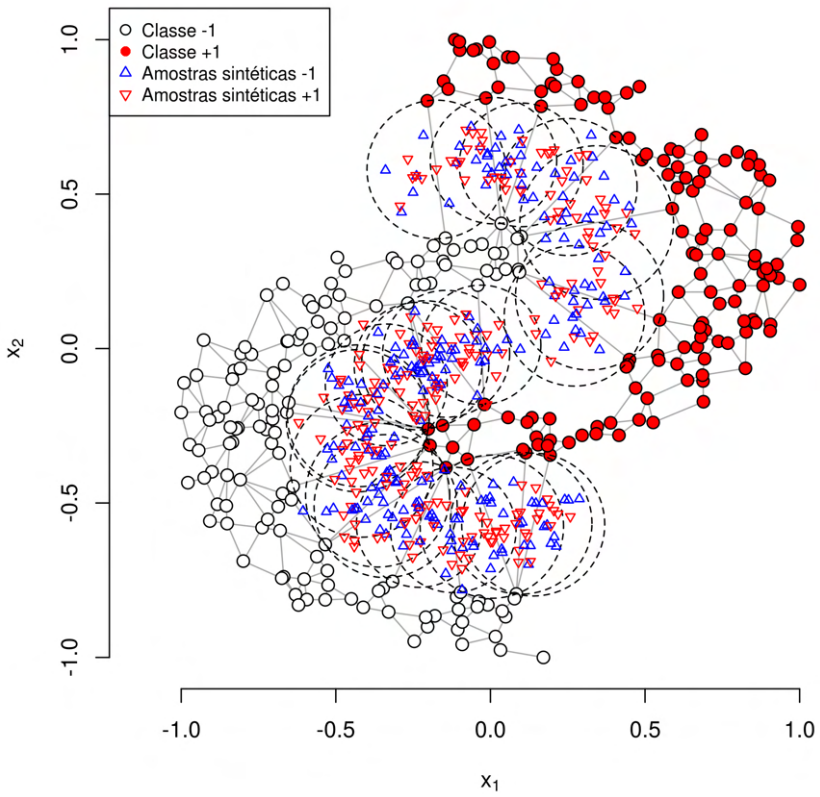


Figura 3.2: Grafo de Gabriel na base de dados *two moons* com amostras sintéticas geradas nas hiperesferas da superfície de separação.

e as arestas definidas conforme a Equação 2.12. O conjunto de treinamento inicial é representado por duas classes, uma com círculo vazio na cor preta (-1) e a outra com círculo cheio na cor vermelha $(+1)$. As amostras sintéticas ruidosas estão próximas da superfície de separação e são representadas por um triângulo azul, $\triangle$, da classe (-1), e por um triângulo invertido vermelho, $\triangledown$, da classe $(+1)$.

Com a adição das amostras sintéticas no treinamento a função de custo Equação 3.1 será atualizada para Equação 3.2.

$$J_e = \sum_{i=1}^{N_1}(y_i - f(\mathbf{h}_i, \mathbf{w}))^2 + \sum_{k=1}^{N_2}(v_k - f(\mathbf{r}_k + \epsilon_k, \mathbf{w}))^2 \tag{3.2}$$

onde N_2 é o número de amostras sintéticas adicionadas no conjunto de treinamento, o termo $(\mathbf{r}_k + \epsilon_k)$ refere-se a k-th amostra sintética, sendo $\mathbf{r}_k$ um ponto de referência definido a partir das informações geométricas do grafo, formado pelo conjunto de treinamento, e ϵ_k um termo de ruído aleatório em relação ao ponto de referência. O termo v_k é o rótulo atribuído a amostra sintética.

O termo adicional na Equação 3.2 causa, portanto, um deslocamento da solução em

direção às amostras $(\mathbf{r}_k + \epsilon_k)$. Assim, quanto maior N_2 maior será a importância das amostras sintéticas na minimização do erro total. Considera-se que as amostras sintéticas estão na região de separação entre os vetores de borda. Nesta região a função discriminante encontra-se na região de transição entre os valores discretos correspondentes às duas classes, que serão aproximadas por uma função linear apresentada pela ELM, de forma que $f(\mathbf{h}_i, \mathbf{w}) \approx \mathbf{w}^T\mathbf{h}_i$ e $f(\mathbf{r}_k + \epsilon_k, \mathbf{w}) \approx \mathbf{w}^T(\mathbf{r}_k + \epsilon_k)$. Substituindo as funções lineares que são uma aproximação da amostra no espaço de características e os pesos da camada de saída na Equação 3.2, temos a Equação 3.3.

$$J_e = \frac{1}{2}\sum_{i=1}^{N_1}(y_i - \mathbf{w}^T\mathbf{h}_i)^2 + \frac{1}{2}\sum_{k=1}^{N_2}(v_k - (\mathbf{w}^T\mathbf{r}_k + \mathbf{w}^T\epsilon_k))^2 \tag{3.3}$$

Considerando-se que $\hat{y}_i = \mathbf{w}^T\mathbf{h}_i$, $\hat{v}_k = \mathbf{w}^T\mathbf{r}_k$ e $\hat{u}_k = \mathbf{w}^T\epsilon_k$ a Equação 3.3 pode ser reescrita na forma da Equação 3.4, em que $\hat{y}_i$ é a resposta do modelo para a amostra de treinamento $\mathbf{x}_i$, $\hat{v}_k$ é a resposta do modelo para a amostra de referência $\mathbf{r}_k$ e $\hat{u}_k$ é a resposta do modelo para o vetor de ruídos adicionado a $\mathbf{r}_k$.

$$J_e = \frac{1}{2}\sum_{i=1}^{N_1}(y_i - \hat{y}_i)^2 + \frac{1}{2}\sum_{k=1}^{N_2}(v_k - (\hat{v}_k + \hat{u}_k))^2 \tag{3.4}$$

Para encontrar o separador $\mathbf{w}$ que minimiza a Equação 3.3, a mesma deve ser diferenciada em relação w_j, a derivada deve ser igual a 0 (zero) assim temos a equação normal para o problema de mínimos quadrados (Bishop, 1995a). O desenvolvimento leva à Equação 3.5 e as equações seguintes.

$$\begin{aligned}\frac{\partial J_e}{\partial w_j} = \; & -\sum_{i=1}^{N_1} y_i h_{ij} - \sum_{k=1}^{N_2} v_k \epsilon_{kj} - \sum_{k=1}^{N_2} v_k r_{kj} \\ & + \sum_{i=1}^{N_1} \hat{y}_i h_{ij} + \sum_{k=1}^{N_2} \hat{v}_k r_{kj} + \sum_{k=1}^{N_2} \hat{u}_k \epsilon_{kj} \\ & + \sum_{k=1}^{N_2} \hat{u}_k r_{kj} + \sum_{k=1}^{N_2} \hat{v}_k \epsilon_{kj}\end{aligned} \tag{3.5}$$

Igualando-se a Equação 3.5 a zero temos a Equação 3.6 .

$$\begin{aligned}& -\sum_{i=1}^{N_1} y_i h_{ij} - \sum_{k=1}^{N_2} v_k \epsilon_{kj} - \sum_{k=1}^{N_2} v_k r_{kj} \\ & + \sum_{i=1}^{N_1} \hat{y}_i h_{ij} + \sum_{k=1}^{N_2} \hat{v}_k r_{kj} + \sum_{k=1}^{N_2} \hat{u}_k \epsilon_{kj} \\ & + \sum_{k=1}^{N_2} \hat{u}_k r_{kj} + \sum_{k=1}^{N_2} \hat{v}_k \epsilon_{kj} = 0\end{aligned} \tag{3.6}$$

A Equação 3.6 pode ser reescrita na forma a seguir,

$$\begin{aligned}& \sum_{i=1}^{N_1} \hat{y}_i h_{ij} + \sum_{k=1}^{N_2} \hat{v}_k r_{kj} + \sum_{k=1}^{N_2} \hat{u}_k \epsilon_{kj} \\ & + \sum_{k=1}^{N_2} \hat{u}_k r_{kj} + \sum_{k=1}^{N_2} \hat{v}_k \epsilon_{kj} = \\ & \sum_{i=1}^{N_1} y_i h_{ij} + \sum_{k=1}^{N_2} v_k r_{kj} + \sum_{k=1}^{N_2} v_k \epsilon_{kj}\end{aligned} \tag{3.7}$$

$$\begin{aligned}& \sum_{i=1}^{N_1} \mathbf{w}^T\mathbf{h}_i h_{ij} + \sum_{k=1}^{N_2} \mathbf{w}^T\mathbf{r}_k r_{kj} + \sum_{k=1}^{N_2} \mathbf{w}^T\epsilon_k \epsilon_{kj} \\ & + \sum_{k=1}^{N_2} \mathbf{w}^T\epsilon_k r_{kj} + \sum_{k=1}^{N_2} \mathbf{w}^T\mathbf{r}_k \, \epsilon_{kj} = \\ & \sum_{i=1}^{N_1} y_i h_{ij} + \sum_{k=1}^{N_2} v_k r_{kj} + \sum_{k=1}^{N_2} v_k \epsilon_{kj}\end{aligned} \tag{3.8}$$

Para facilitar o desenvolvimento da Equação 3.8 a mesma será reescrita na notação matricial, resultam-se nas seguintes equações.

$$\begin{aligned}&\mathbf{H}^T\mathbf{H}\mathbf{w} + \mathbf{R}^T\mathbf{R}\mathbf{w} + \mathbf{E}^T\mathbf{E}\mathbf{w} + \mathbf{R}^T\mathbf{E}\mathbf{w} + \mathbf{E}^T\mathbf{R}\mathbf{w} = \\ &\mathbf{H}^T\mathbf{y} + \mathbf{R}^T\mathbf{v} + \mathbf{E}^T\mathbf{v}\end{aligned} \tag{3.9}$$

$$(\mathbf{H}^T\mathbf{H} + \mathbf{R}^T\mathbf{R} + \mathbf{E}^T\mathbf{E} + \mathbf{R}^T\mathbf{E} + \mathbf{E}^T\mathbf{R})\mathbf{w} = \mathbf{H}^T\mathbf{y} + (\mathbf{R}^T + \mathbf{E}^T)\mathbf{v} \tag{3.10}$$

$$\mathbf{w} = (\mathbf{H}^T\mathbf{H} + \mathbf{R}^T\mathbf{R} + \mathbf{E}^T\mathbf{E} + \mathbf{R}^T\mathbf{E} + \mathbf{E}^T\mathbf{R})^{-1}(\mathbf{H}^T\mathbf{y} + (\mathbf{R}^T + \mathbf{E}^T)\mathbf{v}) \tag{3.11}$$

As matrizes iniciais do conjunto de treinamento e da saída desejada são representados por $\mathbf{H}$ e $\mathbf{y}$ respectivamente,

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} & \dots & h_{1L} \\ h_{21} & h_{22} & \dots & h_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ h_{N_1 1} & h_{N_1 2} & \dots & h_{N_1 L} \end{bmatrix} \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{N_1} \end{bmatrix} \tag{3.12}$$

A matriz das amostras sintéticas com ruído é representada por $\mathbf{P}$ na Equação 3.13 que é a soma das matrizes $\mathbf{R}$ (vetor de referência) e $\mathbf{E}$ (ruído aleatório), representadas respectivamente nas Equações 3.14 e 3.15. A matriz de rótulos das amostras com ruído é dado por $\mathbf{v}$ na Equação 3.16.

$$\mathbf{P} = \mathbf{R} + \mathbf{E} \tag{3.13}$$

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & \dots & r_{1L} \\ r_{21} & r_{22} & \dots & r_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ r_{N_2 1} & r_{N_2 2} & \dots & r_{N_2 L} \end{bmatrix} \tag{3.14}$$

$$\mathbf{E} = \begin{bmatrix} \epsilon_{11} & \epsilon_{12} & \dots & \epsilon_{1L} \\ \epsilon_{21} & \epsilon_{22} & \dots & \epsilon_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ \epsilon_{N_2 1} & \epsilon_{N_2 2} & \dots & \epsilon_{N_2 L} \end{bmatrix} \tag{3.15}$$

$$\mathbf{v} = \begin{bmatrix} v_1 \\ v_2 \\ \vdots \\ v_{N_2} \end{bmatrix} \tag{3.16}$$

A matriz Λ na Equação 3.17 representa o termo de regularização

$$\Lambda = \mathbf{R}^T\mathbf{R} + \mathbf{E}^T\mathbf{E} + \mathbf{R}^T\mathbf{E} + \mathbf{E}^T\mathbf{R} \tag{3.17}$$

Assim, ao se substituir a Equação 3.17 na Equação 3.11 temos a nova Equação 3.18 de atualização dos pesos.

$$\mathbf{w} = (\mathbf{H}^T\mathbf{H} + \Lambda)^{-1}(\mathbf{H}^T\mathbf{y} + (\mathbf{R}^T + \mathbf{E}^T)\mathbf{v}) \tag{3.18}$$

A Equação 3.18 é a solução proposta para ajuste dos pesos do modelo quando a reamostragem é aplicada ao conjunto de treinamento. O termo de regularização Λ, Equação 3.17, é composto pelos termos de reamostragem $\mathbf{R}$ e $\mathbf{E}$ que produzem o efeito de suavização na superfície de separação. Na ausência de ruídos os termos $\mathbf{R}$ e $\mathbf{E}$ são nulos e a Equação 3.18 é reduzida a equação padrão dos mínimos quadrados, Equação 2.17.

A reamostragem local nas hiperesferas esferas foram realizadas de forma balanceada, ou seja, o número de amostras sintéticas da classe -1 e $+1$ são iguais. Será apresentado o efeito esperado das amostras sintéticas ruidosas na suavização da superfície de separação. Portanto, suponha que $N = N_p + N_n$, onde N_p é o número de amostras com rótulo positivo e N_n o número de amostras com rótulo negativo.

$$\sum_{k=1}^{N}(r_k + \epsilon_k)v_k = \sum_{k=1}^{N_p}(r_{pk} + \epsilon_{pk})(+1) + \sum_{k=1}^{N_n}(r_{nk} + \epsilon_{nk})(-1) \tag{3.19}$$

A referência, r, para geração das amostras em cada hiperesfera é única. Como o número de amostras gerado de cada classe é o mesmo, então temos que $r_{pk} = r_{nk} = r$ é constante.

$$\sum_{k=1}^{N}(r_k + \epsilon_k)v_k = N_p r + \sum_{k=1}^{N_p}\epsilon_{pk} - N_n r - \sum_{k=1}^{N_n}\epsilon_{nk} \tag{3.20}$$

Se o conjunto de amostras é balanceado, ou seja, $N_p = N_r = M$:

$$\sum_{k=1}^{N}(r_k + \epsilon_k)v_k = \sum_{k=1}^{M}\epsilon_{pk} - \sum_{k=1}^{M}\epsilon_{nk} = \sum_{k=1}^{M}(\epsilon_{pk} - \epsilon_{nk}) \tag{3.21}$$

Se as amostras sintéticas foram geradas por uma distribuição gaussiana, $\epsilon_p \sim \mathcal{N}(\mu = 0, \sigma^2)$ e $\epsilon_n \sim \mathcal{N}(\mu = 0, \sigma^2)$, $\epsilon_d = (\epsilon_p - \epsilon_n) \sim \mathcal{N}(\mu = 0, 2\sigma^2)$,

$$\sum_{k=1}^{N}(r_k + \epsilon_k)v_k = \sum_{k=1}^{M}\epsilon_{dk} \tag{3.22}$$

Finalmente de acordo com a Lei dos Grandes Números (DeGroot & Schervish, 2012), como as variáveis ϵ_d são independente e identicamente distribuídas , quando $M \to \infty$, $\frac{1}{M}\sum_{k=1}^{M}\epsilon_{dk}$ converge provavelmente para μ,

$$\sum_{k=1}^{N}(r_k + \epsilon_k)v_k = \sum_{k=1}^{M}\epsilon_{dk} = M\mu = M(0) = 0 \tag{3.23}$$

O resultado apresentado na Equação 3.23 prova que, para uma quantidade suficientemente grande de amostras geradas, o termo $(\mathbf{R}^T + \mathbf{E}^T)\mathbf{v}$ na Equação 3.18 é igual a zero, desta forma a Equação 3.18 pode ser reescrita como a Equação 3.24.

$$\mathbf{w} = (\mathbf{H}^T\mathbf{H} + \Lambda)^{-1}(\mathbf{H}^T\mathbf{y}) \tag{3.24}$$

Analisando a Equação 3.24, pode ser visto que a adição suficiente de amostras sintéticas é equivalente a regularização de Tikhonov.

A fim de evitar a necessidade de geração de um número assintoticamente grande de amostras, a simetria das amostras é realizada. A simetria é obtida desde que cada amostra seja espelhada em relação a r e tenha o mesmo rótulo. As Equações 3.25 e 3.26 apresentam a simetria das amostras.

$$\mathbf{P}' = \mathbf{R} - \mathbf{E} \tag{3.25}$$

$$\mathbf{v}' = \mathbf{v} \tag{3.26}$$

Para o conjunto sintético temos então a Equação 3.27.

$$\mathbf{P}^T\mathbf{v} + \mathbf{P}'^T\mathbf{v}' = 0 \tag{3.27}$$

Esta estratégia tem o objetivo de fazer com que a soma de todas as amostras sintéticas geradas seja zero para que não ocorra interferência com as características da base de dados. Portanto, as amostras ruidosas geradas influenciam somente no efeito de regularização.

3.1.3 Conclusão do capítulo

O método proposto neste trabalho tem como objetivo reduzir a complexidade da rede neural para evitar *overfitting* de uma rede neural superdimensionada em relação ao problema. A ideia de adição de amostras sintéticas no treinamento da rede neural foi baseado no treinamento com adição de ruídos que tem como efeito a regularização de Tikhonov. A abordagem deste trabalho é realizar reamostragem na margem de separação em um espaço de alta dimensionalidade onde a classes podem ser linearmente separáveis. A partir do desenvolvimento matemático da função de custo com a adição das amostras sintéticas no treinamento foi possível observar que o método proposto apresenta o mesmo efeito da regularização de Tikhonov.

Para a suavização da superfície de separação as amostras sintéticas adicionadas no conjunto de treinamento foram balanceadas sendo o mesmo número de amostras para

ambas as classes e com amostras simétricas dentro da hiperesfera na margem de separação. Pela lei dos Grandes Números vimos que para um grande número de amostras temos que as amostras sintéticas convergem para a região mais provável de maximização da margem de separação.

Capítulo 4

Experimentos e resultados

Este capítulo descreve a avaliação do modelo de adição de amostras sintéticas proposto no capítulo 3 chamado RN-ELM. Dois conjuntos de bases de dados foram utilizados: 4 bases de dados sintéticas bidimensionais e 18 bases de dados reais.

Nos experimentos com bases de dados sintéticas foram comparados os métodos ELM padrão (Huang *et al.*, 2006) e o método proposto neste trabalho (RN-ELM). Nas bases dimensionais é possível visualizar a superfície de separação e os efeitos dos modelos aplicados. Nos experimentos com bases de dados reais para comparar com o modelo RN-ELM, foram utilizados dois modelos: o ELM padrão (Huang *et al.*, 2006) e ELM-REG (Huang *et al.*, 2012; Silvestre, 2015) que acrescenta um termo de regularização no cálculo da pseudo-inversa.

Os modelos ELM, ELM-REG e RN-ELM foram comparados em redes neurais com grandes números de neurônios na camada oculta. A acurácia e a norma dos pesos da rede serão utilizados para comparação dos modelos. A primeira em relação a capacidade de generalização dos modelos e a segunda em relação ao ajuste da complexidade do modelo ao problema proposto. A influência da regularização serão observadas no modelo proposto, RN-ELM, e no modelo ELM-REG. Para verificar se existem diferenças significativas entre os modelos serão utilizados o teste não paramétrico de Friedman e o teste *post hoc* de Nemenyi para comparação dos modelos dois a dois.

4.1 Metodologia

Foram avaliados problemas de classificação binária em que os dados de entrada foram pré-processados da seguinte forma: as amostras que possuem dados ausentes foram removidas do conjunto de dados, foi feita a padronização dos dados com média 0 e desvio padrão 1, e por fim, o rótulo das amostras foram ajustados para -1 ou +1. A rede neural ELM foi escolhida devido a simplicidade de configuração e por possuir apenas uma camada oculta. Desta forma, somente os pesos da camada de saída são ajustados no treinamento e assim, sua rapidez por não utilizar métodos iterativos.

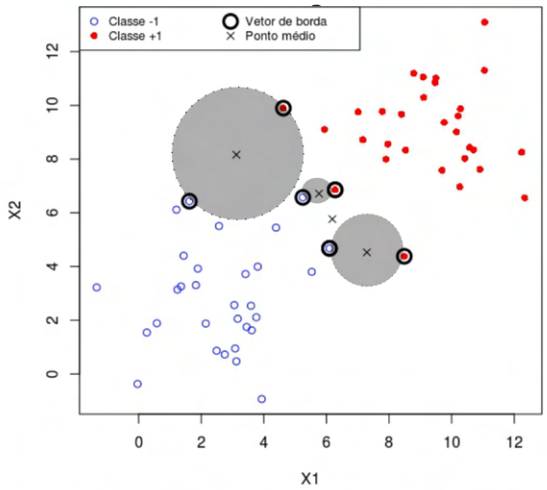
Segundo o Teorema de Cover (1965), um problema não linearmente separável quando projetado em um espaço de alta dimensão tem a probabilidade de ser linearmente separável. Assim, ao projetar as bases de dados em um espaço de alta dimensionalidade em uma rede neural de maior complexidade com grande número de neurônios na camada oculta, o modelo deve ser capaz de configurar a rede neural para a complexidade do problema e desta maneira, melhorar a capacidade de generalização e evitar *overfitting*.

A configuração geral das redes neurais foram assim definidas. Os pesos da camada de entrada foram selecionados a partir de uma distribuição uniforme no intervalo de $-0,5$ até $+0,5$, a função de ativação utilizada na camada oculta foi a tangente hiperbólica, o número de neurônios da camada oculta foram definidos como 10, 30, 100, 500 e 1000 (Silvestre *et al.*, 2015). Para o modelo ELM-REG o parâmetro de regularização C foi selecionado no intervalo de 2^{-24} até 2^{25} conforme (Huang *et al.*, 2012; Silvestre *et al.*, 2015).

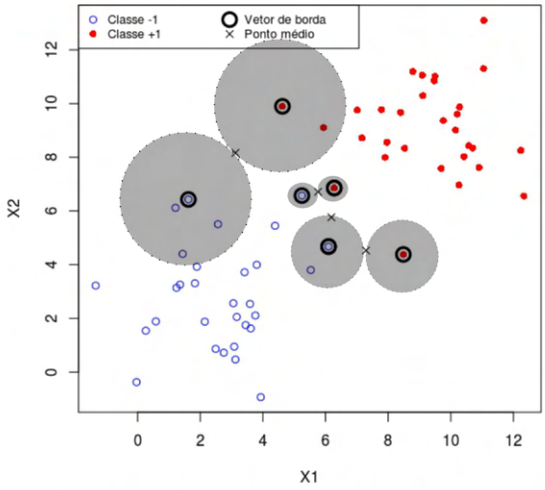
4.1.1 Determinar a região dentro da margem para adição de ruídos

Os primeiros experimentos realizados foram para determinar a região dentro da margem de separação para adição das amostras sintéticas ruidosas. Para identificar a melhor região de ajuste do modelo foram adicionadas amostras sintéticas ruidosas nas seguintes regiões: no entorno do ponto médio Figura 4.1(a), no entorno dos vetores de borda Figura 4.1(b), entre o ponto médio e o vetor de borda Figura 4.1(c) e entre vetores de borda Figura 4.1(d). As amostras simétricas foram avaliadas nesta etapa e as métricas utilizadas foram o erro quadrático médio e a norma dos pesos.

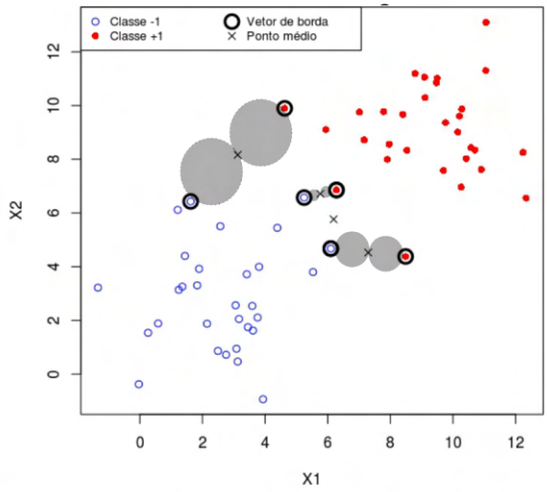
O resultado indicou que a região entorno do ponto médio diminuiu a complexidade das rede com adição das amostras sintéticas com simetria. Nos demais casos ocorreu aumento da norma dos pesos da rede com adição das amostras sintéticas. Os experimentos foram descritos no Apêndice A.



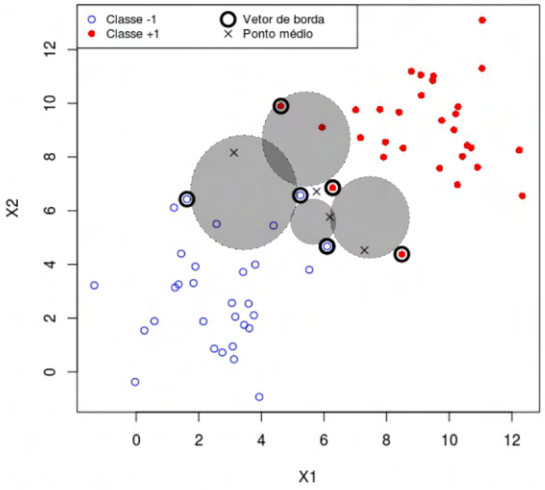
(a) Região de adição de ruídos no entorno do ponto médio da hiperesfera.



(b) Região de adição de ruídos no entorno do vetor de borda.



(c) Região de adição de ruídos entre vetores de borda e ponto médio.



(d) Região de adição de ruídos entre vetores de borda.

4.2 Descrição das bases de dados

As bases de dados utilizadas nos experimentos deste trabalho são formadas por bases de dados sintéticas obtidas em (Torres, 2016) e bases de dados reais obtidas nos repositórios UCI (Dheeru & Karra Taniskidou, 2017) e KEEL (Alcala-Fdez *et al.*, 2011).

4.2.1 Bases de dados sintéticas

As bases de dados sintéticas bidimensionais permitem visualizar graficamente a capacidade dos modelos em relação a regularização e a superfície de separação. Os modelos podem ser executados em um conjunto de amostras com sobreposição de classes distintas, e desta forma, a superfície de separação entre as classes não são separáveis linearmente. A complexidade do modelo pode ser calculada segundo a norma dos pesos da rede e o desempenho pela acurácia. O objetivo de utilizar bases bidimensionais é poder visualizar

graficamente a superfície de separação do modelo e comparar o desempenho do modelo proposto RN-ELM que utiliza regularização em relação ao modelo ELM padrão. Neste momento serão comparados os modelos ELM padrão e RN-ELM, com 500 neurônios na camada oculta. A comparação entre os modelos ELM-REG e RN-ELM será realizada utilizando bases de dados reais. A etapa de treinamento com as bases sintéticas utiliza todas as amostras e tem como resultado a superfície de separação apresentada nas figuras dos modelos. Para o modelo proposto foram realizadas 30 repetições para determinação de sua complexidade. Embora o conjunto de treinamento seja o mesmo, as amostras sintéticas adicionadas no treinamento alteram em cada repetição e, assim, a norma dos pesos do modelo RN-ELM foi a média de 30 repetições.

4.2.1.1 Resultados da base de dados sintética *two moons*

Para comparar a capacidade de regularização do modelo proposto RN-ELM com o modelo padrão ELM, em ambos os métodos foi aplicada a base de dados *two moons*. A superfície de separação do modelo ELM com 500 neurônios na camada oculta pode ser visto na Figura 4.1, a acurácia do modelo foi de $94,23\%$ e a complexidade da rede dada pela norma dos pesos foi de $7,62$ x 10^{10}, que resultou em *overfitting*,

A superfície de separação do modelo RN-ELM com o mesmo número de neurônios na camada oculta e com regularização aplicada pela reamostragem local pode ser vista na Figura 4.2. A acurácia do modelo foi de $94,87\%$ e a norma dos pesos do modelo foi de $2,85$x10^4. Nesta comparação não houve diferença significativa na acurácia entre os modelos. No entanto, houve redução da complexidade da rede neural pela redução da norma dos pesos em seis ordens de grandezas.

4.2.1.2 Resultado da base de dados sintética *half kernel*

A rede neural ELM com 500 neurônios na camada oculta e o treinamento utilizam todas as amostras e tem como resultado a superfície de separação. Ambos os modelos ELM padrão e o RN-ELM foram capazes de definir uma superfície de separação que divide as duas classes. O modelo ELM padrão tem a norma dos pesos de $1,00$x10^6 e a superfície de separação pode ser vista na Figura 4.3. A norma dos pesos do modelo RN-ELM foi de $78,24$ e a superfície de separação pode ser vista na Figura 4.4. Com a comparação das duas superfícies foi possível observar que o modelo proposto foi capaz de reduzir a complexidade da rede neural e suavizar a superfície em uma rede neural para o problema proposto.

4.2.1.3 Resultados da base de dados sintética *corners*

Ambos os modelos ELM padrão e o RN-ELM foram capazes de definir uma superfície de separação que divide as duas classes. A superfície de separação da rede neural ELM

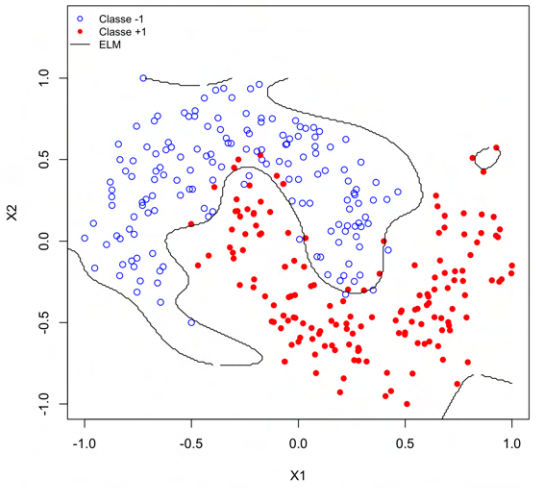


Figura 4.1: Superfície de separação do modelo ELM padrão da base de dados *two moon* com 500 neurônios na camada oculta com overfitting.

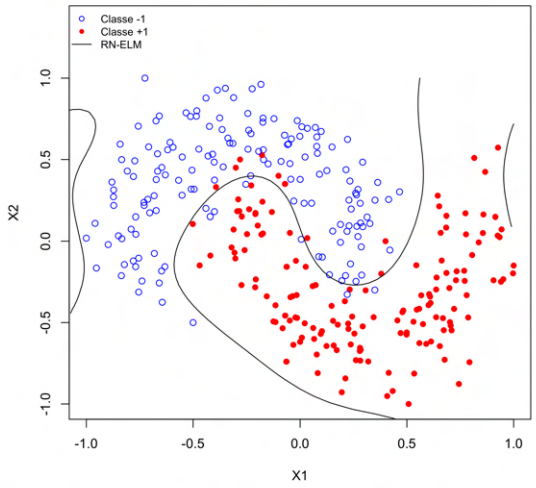


Figura 4.2: Superfície de separação do modelo RN-ELM da base de dados *two moon* com 500 neurônios na camada oculta com suavização.

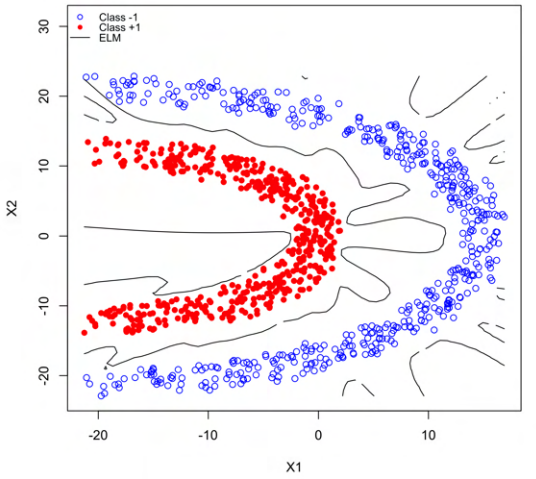


Figura 4.3: Superfície de separação do modelo ELM padrão da base de dados *half kernel* com 500 neurônios na camada oculta com overfitting.

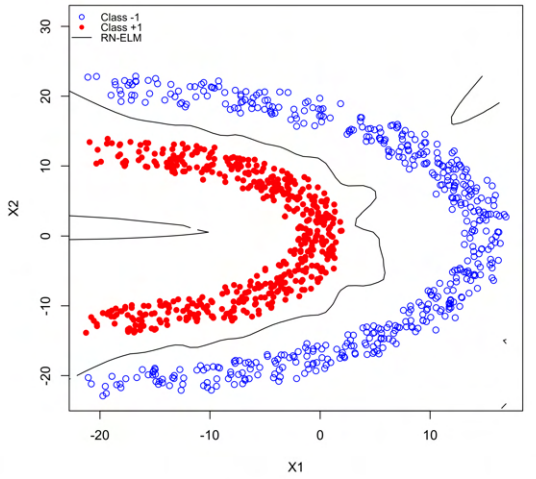


Figura 4.4: Superfície de separação do modelo RN-ELM da base de dados *half kernel* com 500 neurônios na camada oculta com suavização.

padrão com 500 neurônios na camada oculta é apresentada na Figura 4.5 e obteve a norma dos pesos de $4,87\text{x}10^5$. A rede neural artificial com 500 neurônios que utiliza o modelo RN-ELM tem a superfície de separação apresentada na Figura 4.6 e obteve a norma dos pesos de $146,67$. O modelo RN-ELM foi capaz de reduzir a complexidade da rede para se adaptar ao problema de classificação.

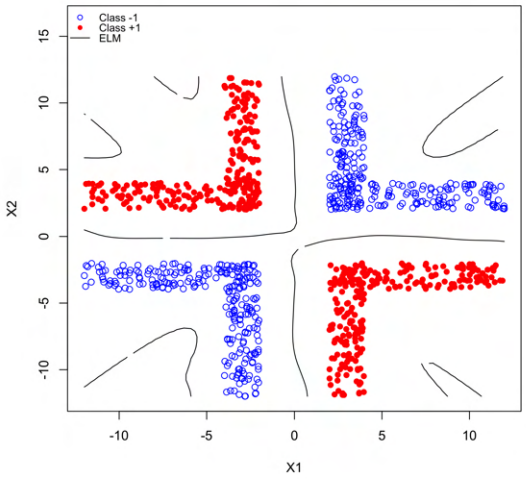


Figura 4.5: Superfície de separação do modelo ELM padrão da base de dados *corners* com 500 neurônios na camada oculta com overfitting.

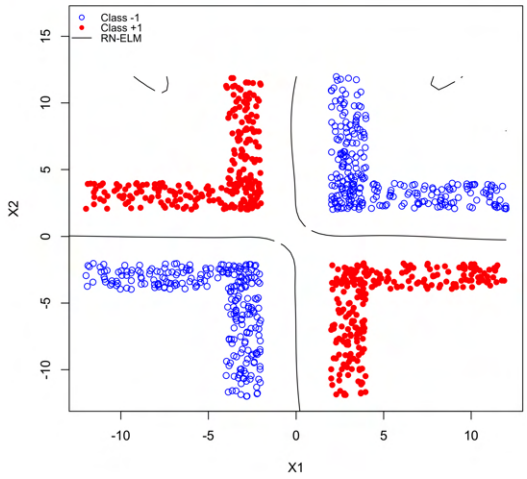


Figura 4.6: Superfície de separação do modelo RN-ELM da base de dados *corners* com 500 neurônios na camada oculta com suavização.

4.2.1.4 Resultado da base de dados sintética *cluster in cluster*

Ambos os modelos ELM padrão e o RN-ELM foram capazes de definir uma superfície de separação que divide as duas classes. A superfície de separação do modelo ELM padrão com 500 neurônios na camada oculta pode ser visto na Figura 4.7 e possui a norma dos pesos de $1980,65$. A superfície de separação do modelo RN-ELM com 500 neurônios na camada oculta pode ser visto na Figura 4.8 e a norma dos pesos de $11,29$. Houve redução da complexidade com o treinamento do modelo RN-ELM.

4.2.2 Bases de dados reais

O desempenho do modelo RN-ELM foi comparado em relação aos modelos ELM e ELM-REG em que foram aplicados em bases de dados reais para problemas de classificação binária. Foram avaliadas 18 bases de dados com diferentes números de atributos e número de amostras. Dez bases de dados foram obtidas no repositório UCI (Dheeru & Karra Taniskidou, 2017) (*Audit Data* (aud), *Australian Credit Approval* (aca), *Wisconsin Diagnostic Breast Cancer* (bcr), *Diabetic Retinopathy* (drp), *Ionosphere* (ion), *Parkinsons* (pks), *Pima Indians Diabetes* (pid), *QSAR biodegradation* (qsr), *Sonar* (snr) e *Statlog* (sth))e seis bases de dados foram obtidas do repositório KEEL (Alcala-Fdez *et al.*, 2011) (*Appendicitis* (apd), Bupa (bpa), *Ecoli1* (ec1), *Haberman* (hbm), *Monk2* (mk2), *Breast Cancer Wisconsin Original* (wcs)) e as bases de dados *Gollub* (glb) (Golub *et al.*, 1999b), *Hess* (hes) (Hess *et al.*, 2006). As amostras com dados faltantes foram desconsideradas. As informações das bases de dados reais foram descritas na Tabela 4.1 com o número de atributos, o número de amostras de cada classe entre parenteses e o número total de amostras

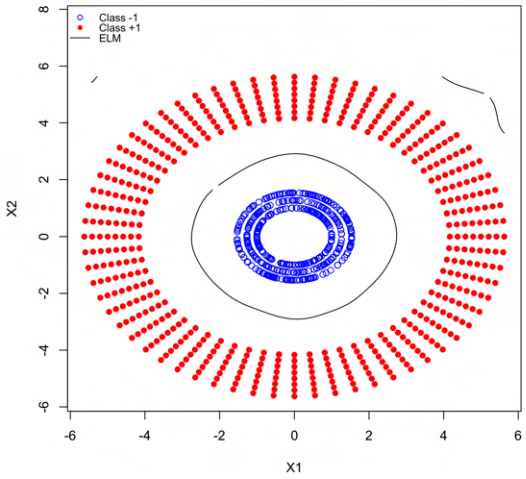


Figura 4.7: Superfície de separação do modelo ELM padrão da base de dados *corners* com 500 neurônios na camada oculta com overfitting.

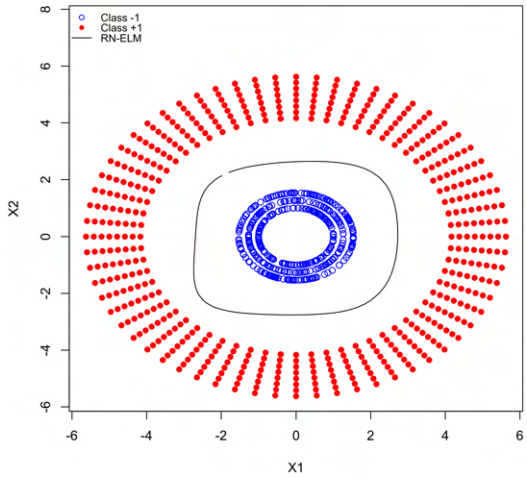


Figura 4.8: Superfície de separação do modelo RN-ELM da base de dados *corners* com 500 neurônios na camada oculta com suavização.

da base. O desempenho do método proposto (RN-ELM) foi comparado com outras duas abordagens do método ELM, uma versão padrão (ELM) e uma versão com regularização no aprendizado (ELM-REG). A seguinte configuração foi feita na rede ELM, as amostras de entrada foram padronizados com média 0 e desvio padrão 1, a saída da rede neural foi rotulada como -1 e +1. Os neurônios da camada oculta utilizaram como função de ativação a tangente hiperbólica e os pesos da camada de entrada foram definidos por uma distribuição uniforme no intervalo de $[-0,5; 0,5]$.

A capacidade de discriminação das rede ELM está relacionada com a capacidade de projetar a rede neural à complexidade do problema (Abu-Mostafa, 1989; Karsoliya & Azad, 2012). Desta forma, para variar a capacidade de discriminação da rede neural, o número de neurônios da camada oculta foi definido por 10, 30, 100, 500 e 1000 (Silvestre *et al.*, 2015). O conjunto de dados foi dividido em conjunto de treinamento e conjunto de testes, na proporção de 70% e 30%, respectivamente. A reamostragem no treinamento da rede neural foi definida por validação cruzada com *10-folds* variando de 1 até 10 amostras em cada ponto médio. A reamostragem local foi definida por uma distribuição normal com desvio padrão dado pela Equação 4.1, que garante que todos os dados estejam a três desvios padrões da média.

$$\sigma = \left(\frac{D}{6}\right) \tag{4.1}$$

onde D é a distância entre os vértices de borda de classes opostas. O parâmetro de regularização (C) da rede neural ELM-REG foi selecionado dentro do intervalo $\{2^{-24} \ldots, 2^{15}\}$ e definido por validação cruzada com *10-folds* (Huang *et al.*, 2012).

Tabela 4.1: Características das bases de dados

Bases de dados	**# de variáveis**	**# de amostras** (-1)(+1) Total
Appendicitis (apd)	8	(85)(21) 106
Audit data (aud)	18	(289)(486) 775
Australian Credit Approval (aca)	15	(383)(307) 690
Wisconsin Diagnostic Breast Cancer (bcr)	31	(357)(212) 569
Bupa (bpa)	7	(145)(200) 345
Diabetic Retinopathy (drp)	20	(540)(611) 1151
Ecoli1 (ec1)	8	(259)(77) 336
Gollub (glb)	51	(25)(47) 72
Haberman (hbm)	4	(225)(81) 306
Hess (hes)	31	(34)(99) 133
Ionosphere (ion)	35	(126)(225) 351
Monk2 (mk2)	7	(204)(228) 432
Parkinsona(pks)	23	(48)(147) 195
Pima Indians Diabetes (pid)	9	(500)(268) 768
QSAR biodegradation (qsr)	42	(699)(356) 1055
Sonar (snr)	61	(111)(97) 208
Statlog (sth)	14	(150)(120) 270
Breast Cancer Wisconsin Original (wcs)	10	(444)(239) 683

4.2.2.1 Resultados das bases de dados reais

O desempenho geral dos métodos de treinamento ELM, ELM-REG e RN-ELM foram comparados pela acurácia média demonstrado na Tabela 4.2 e pela norma dos pesos ($||\mathbf{w}||$) demonstrado na Tabela 4.3. Os valores médios foram resultados de 30 execuções em cada configuração da rede neural. O número de amostras sintéticas (nA) e o número de hiperesferas (nH) geradas pelo modelo RN-ELM pode ser visto na Tabela 4.4 para cada base de dados reais. O total de amostras geradas é dado por $(nA * nH)$. Para comparar os resultados entre os modelos em cada configuração da rede neural $L = \{10, 30, 100, 500$ e $1000\}$ foram realizados testes estatísticos de Friedman (1937, 1940) e o teste *post hoc* de Nemenyi (1963).

A representação das Figuras 4.9, 4.11, 4.13, 4.15, 4.17, 4.19, 4.21, 4.23, 4.25, 4.27, 4.29, 4.31, 4.33, 4.35, 4.39, 4.37, 4.41 e 4.43 mostram o desempenho dos modelos pela acurácia média em relação ao número de neurônios na camada oculta (L). As Figuras 4.10, 4.12, 4.14, 4.16, 4.18, 4.20, 4.22, 4.24, 4.26, 4.28, 4.30, 4.32, 4.34, 4.36, 4.40, 4.38, 4.42 e 4.44 mostram o logaritmo natural da norma dos pesos ($ln(||\mathbf{w}||)$) de cada modelo em função do número de neurônios na camada oculta (L).

Os resultados mostraram que a complexidade do modelo ELM apresentou uma relação direta em relação ao aumento do número de neurônios. O desempenho diminuiu com o aumento da complexidade da rede neural. Este comportamento foi observado em 10 bases de dados representadas nas Figuras: 4.11 e 4.12 (aud), 4.13 e 4.14 (aca), 4.17 e 4.18 (bpa), 4.19 e 4.20 (drp), 4.21 e 4.22 (ec1), 4.25 e 4.26 (hbm), 4.35 e 4.36 (pid), 4.39 e

4.40 (qsr), 4.41 e 4.42 (sth) e 4.43 e 4.44 (wcs). Então, com o aumento do número de neurônios na camada oculta fez com que a norma dos pesos aumentasse e dimimuiu a acurácia do modelo ELM. Nas demais bases de dados os modelos ELM, ELM-REG e RN-ELM obtiveram os mesmos valores para a norma dos pesos com o aumento do número de neurônios na camada oculta, uma relação direta de aumento. Este comportamento foi representado nas Figuras: 4.15 e 4.16 (bcr), 4.23 e 4.24 (glb), 4.27 e 4.28 (hes), 4.29 e 4.30 (ion), 4.33 e 4.34 (pks), 4.37 e 4.38 (snr) e 4.41 e 4.42 (sth). Apesar dos modelos ELM-REG e RN-ELM não reduzirem a complexidade da rede os mesmos tiveram melhor desempenho na maioria das bases de dados. Pode-se sugerir que a projeção das bases de dados no espaço de características não possuem sobreposição, portanto não houve redução da norma dos pesos.

Os modelos ELM-REG e RN-ELM alcançaram melhores resultados de acurácia em relação ao modelo ELM padrão. Os modelos ELM-REG e RN-ELM apresentaram comportamentos semelhantes de desempenho e complexidade da rede com o aumento do número de neurônios na camada oculta. Este comportamento pode ser visto em todas as de dados representadas pelas Figuras de 4.11 até 4.44, o único caso que ocorreu aumento da norma dos pesos da rede RN-ELM e diminuição da acurácia foi na base apd nas Figuras 4.9 e 4.10.

O número de amostras sintéticas ruidosas adicionadas durante o treinamento do modelo RN-ELM nas bases de dados reais estão demonstrados na Tabela 4.4. O número máximo de 40 amostras sintéticas geradas na hiperesfera foi atingido em 4 bases de dados indicadas por apd, bcr, ion e snr. A base de dados apd com L=10 a acurácia da ELM padrão foi maior que a acurácia da RN-ELM.

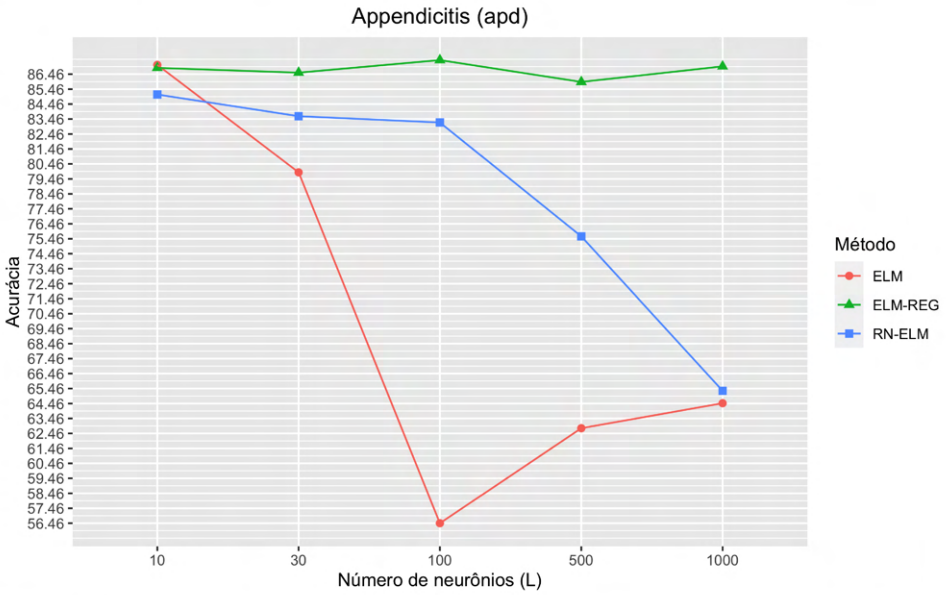


Figura 4.9: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

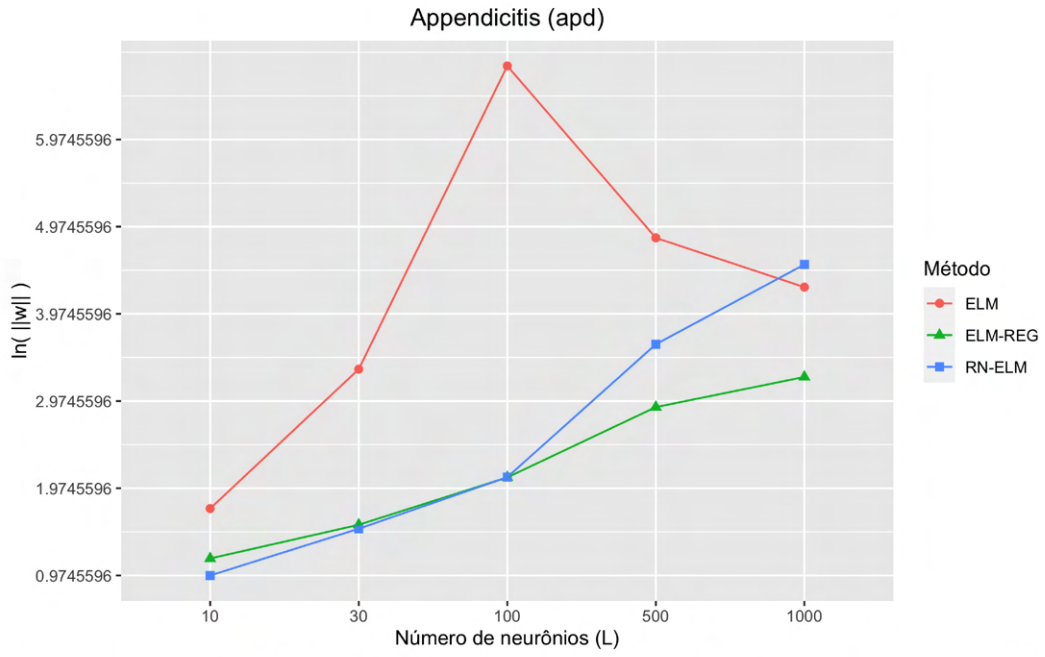


Figura 4.10: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

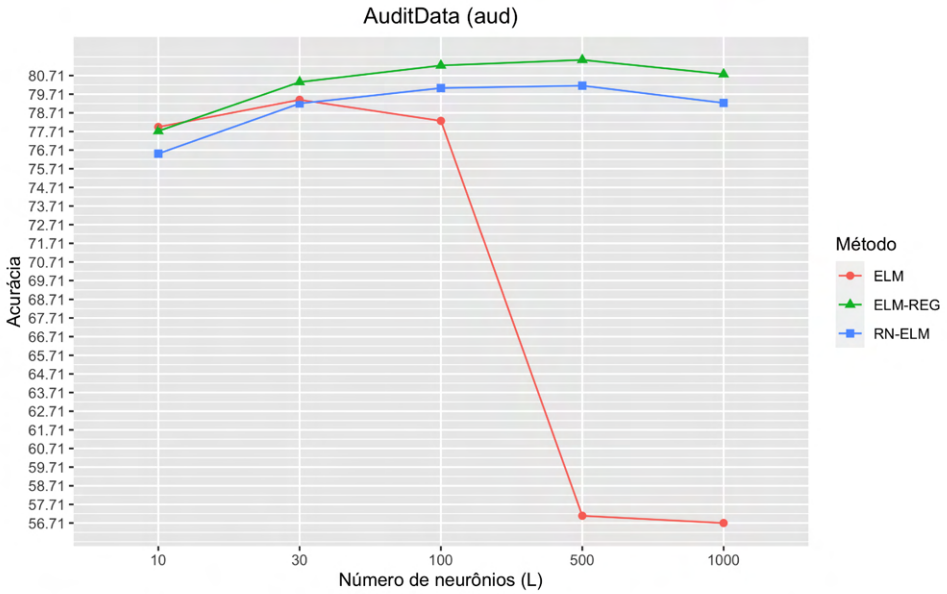


Figura 4.11: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

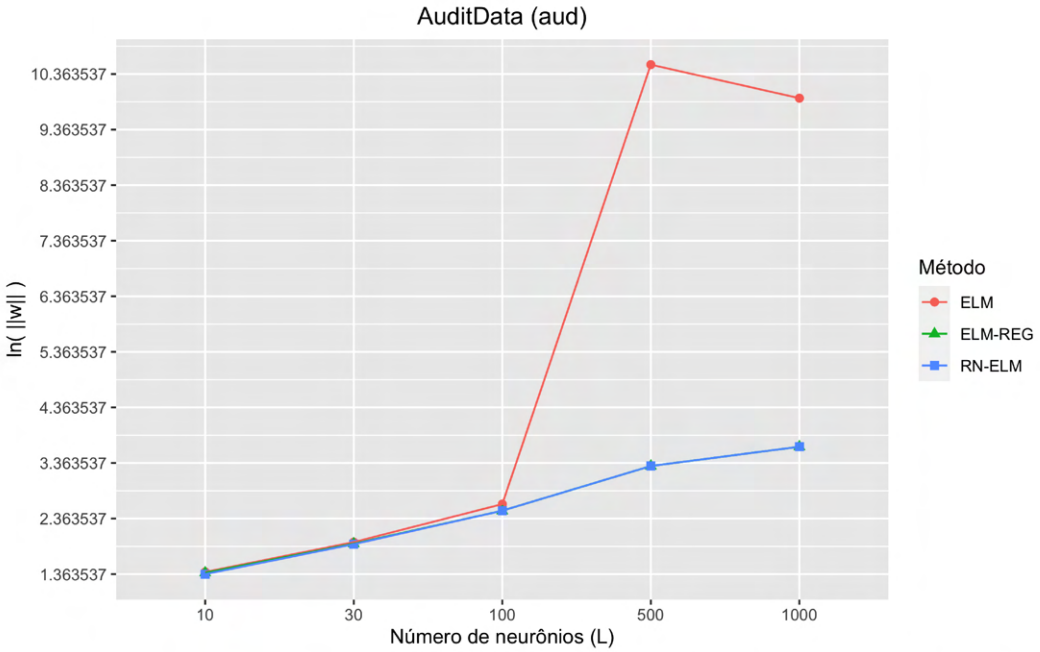


Figura 4.12: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

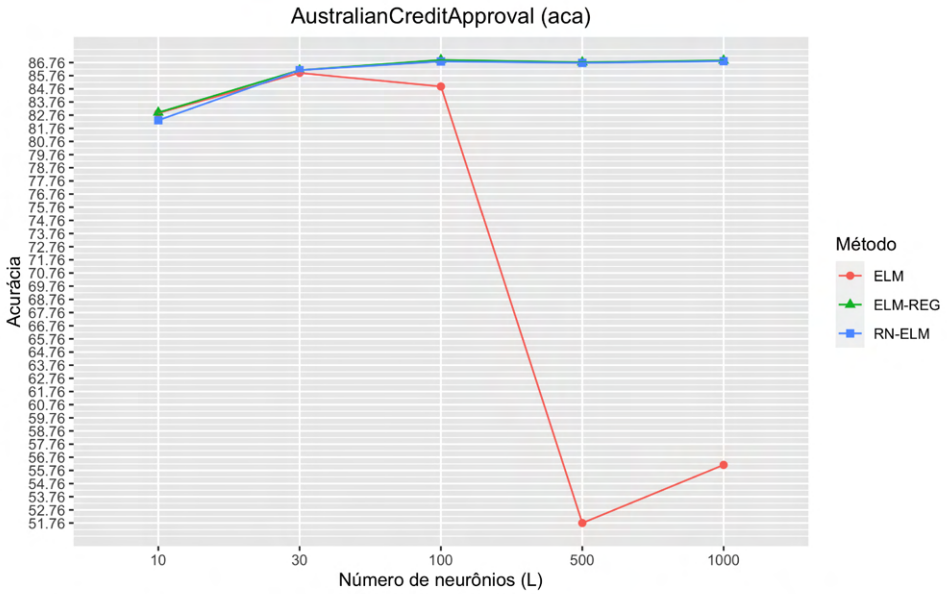


Figura 4.13: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

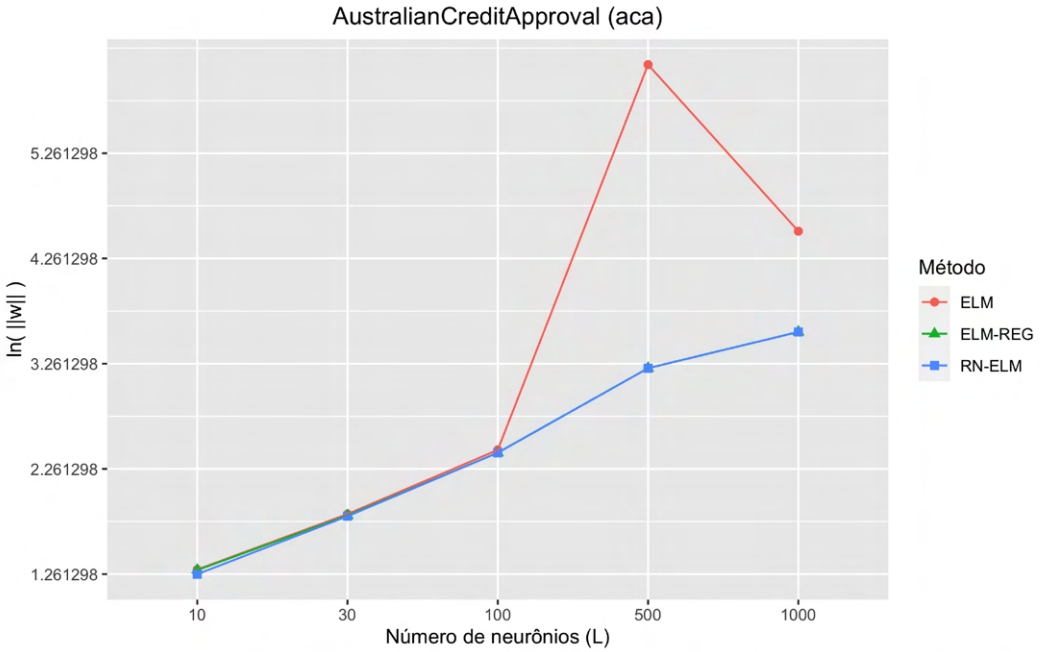


Figura 4.14: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

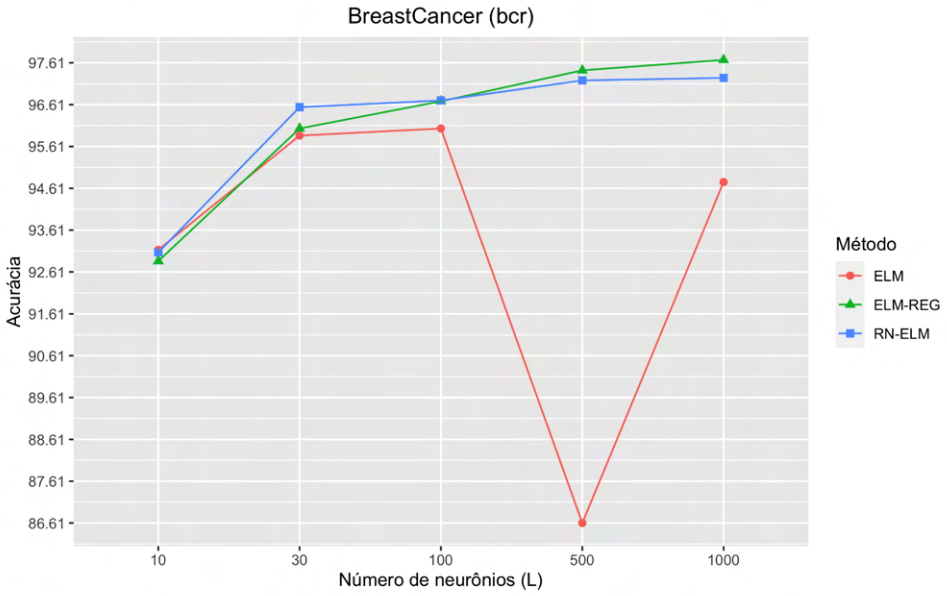


Figura 4.15: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

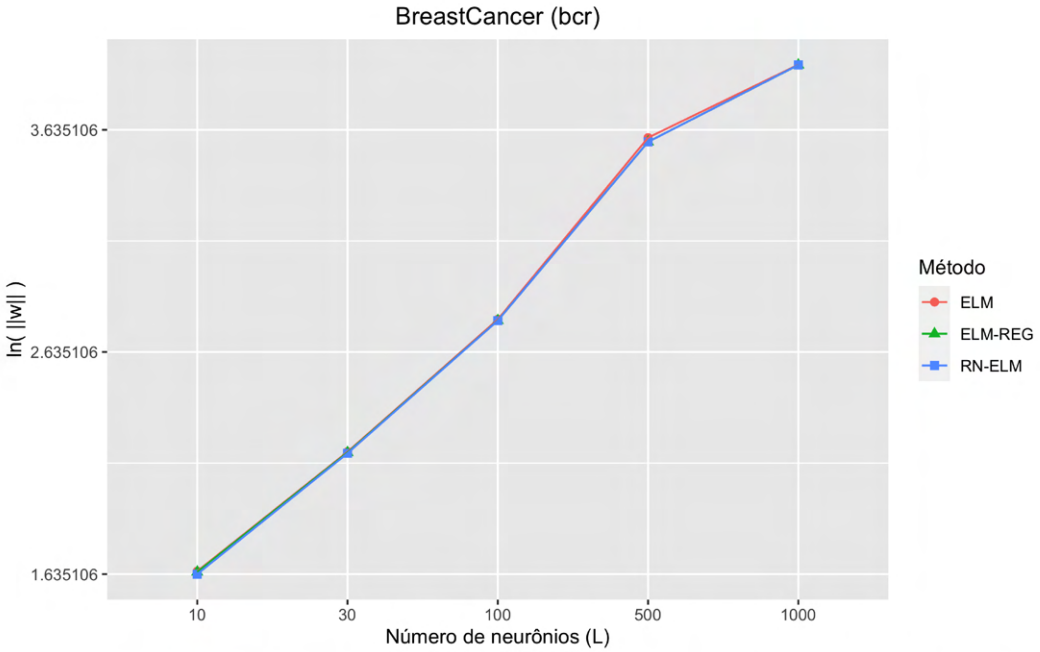


Figura 4.16: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

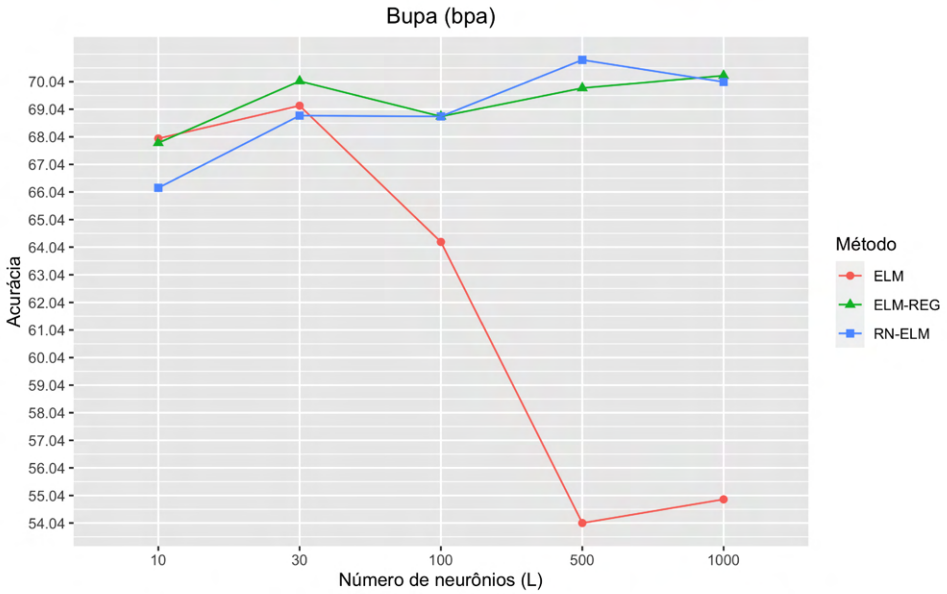


Figura 4.17: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

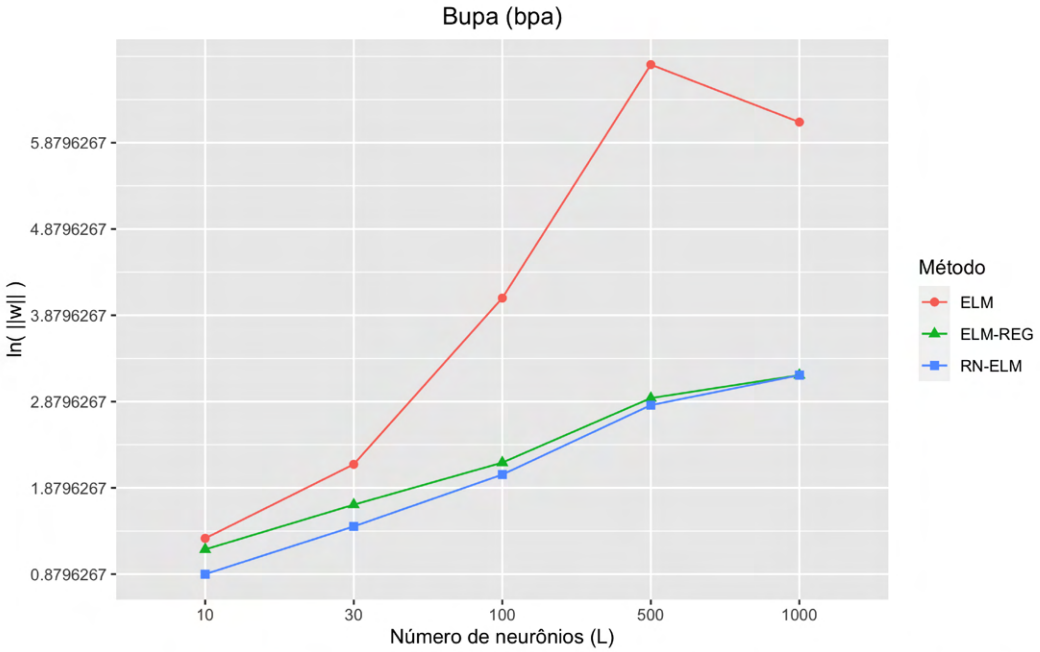


Figura 4.18: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

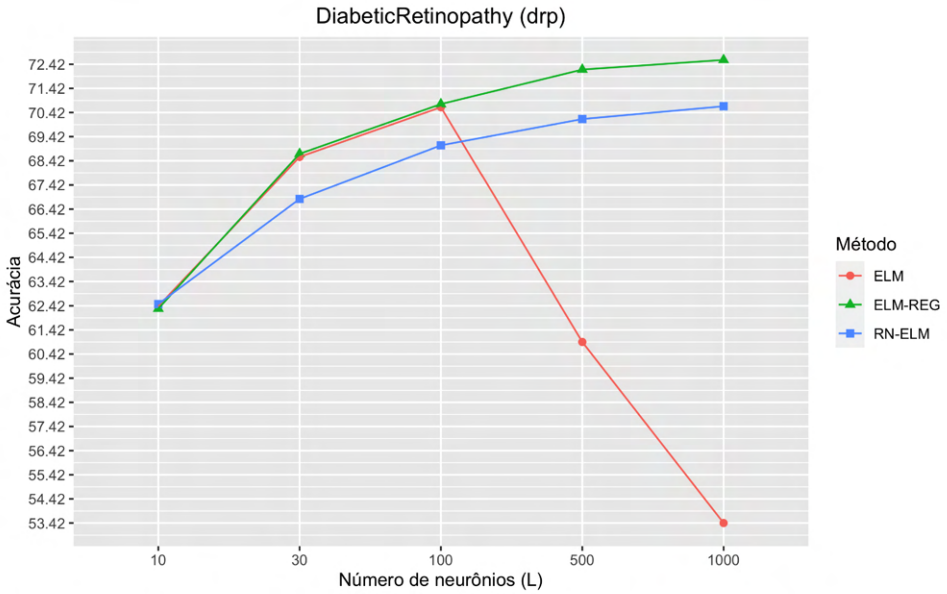


Figura 4.19: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

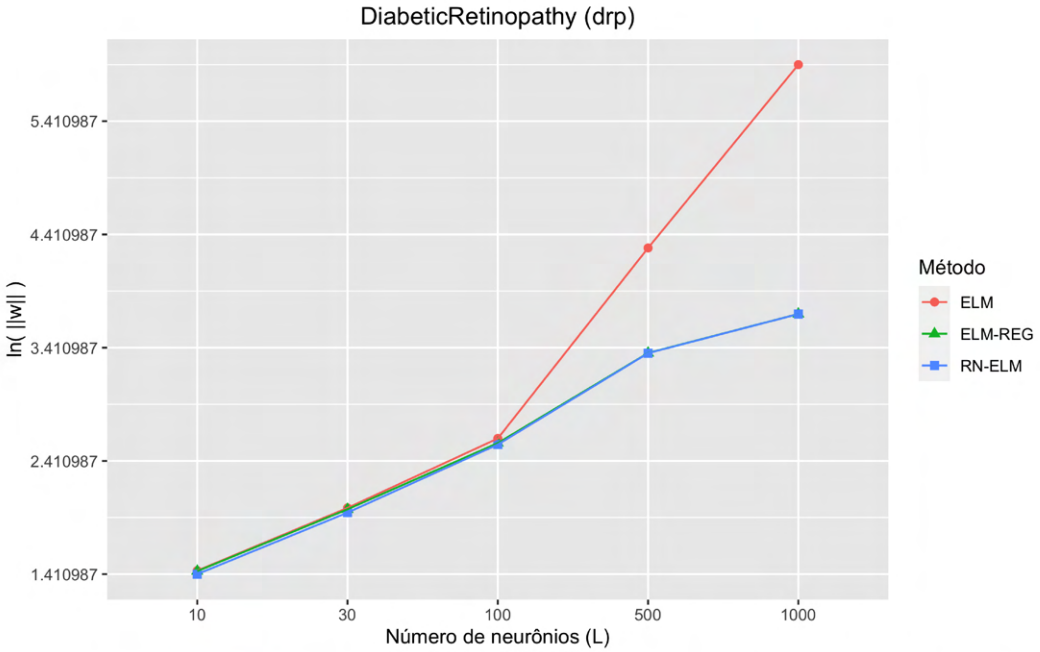


Figura 4.20: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

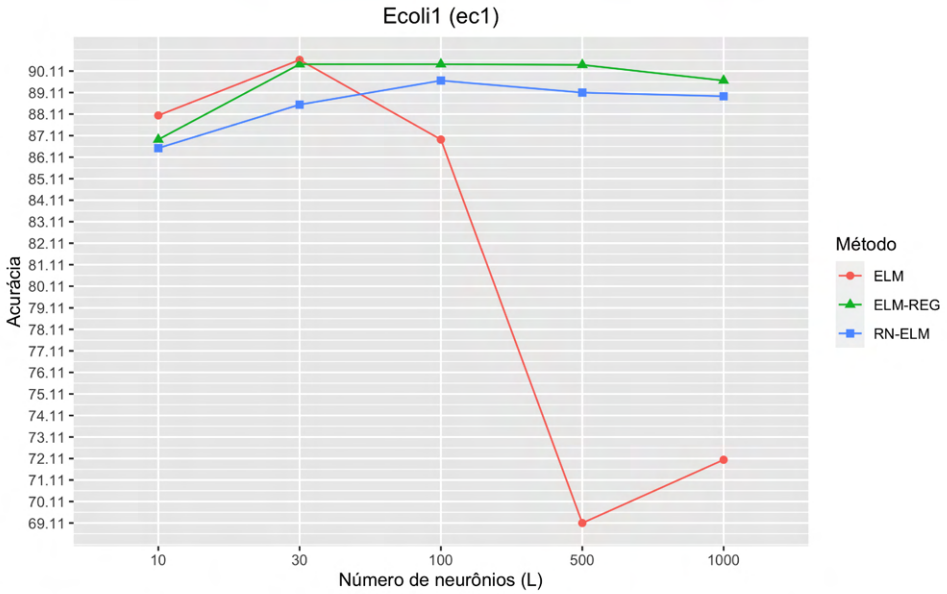


Figura 4.21: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

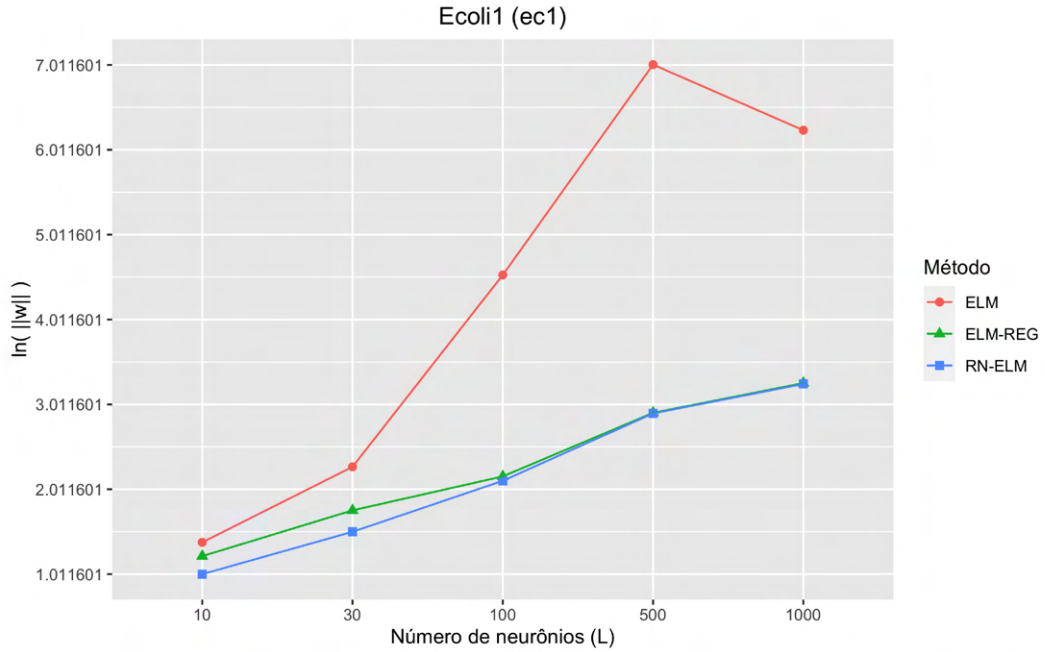


Figura 4.22: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

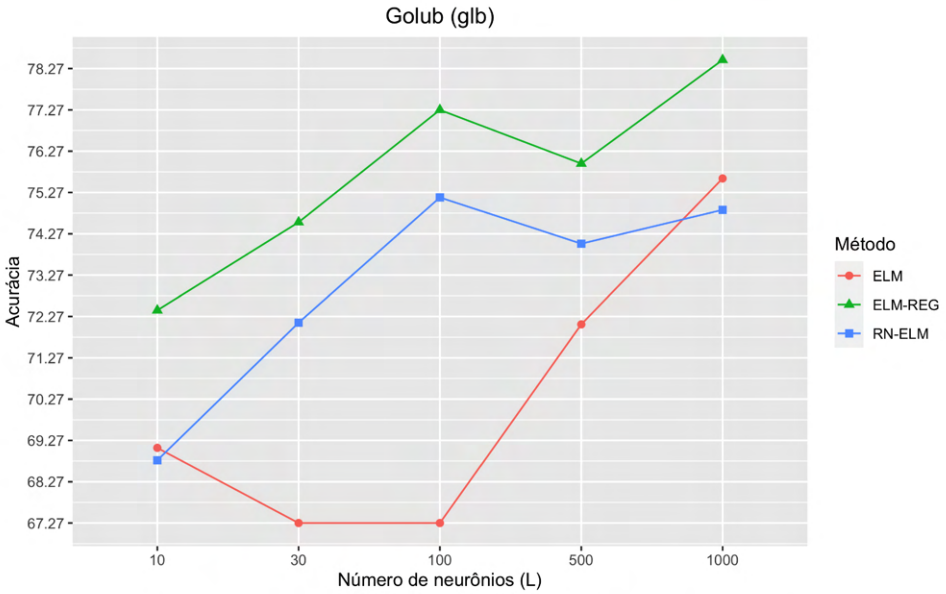


Figura 4.23: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

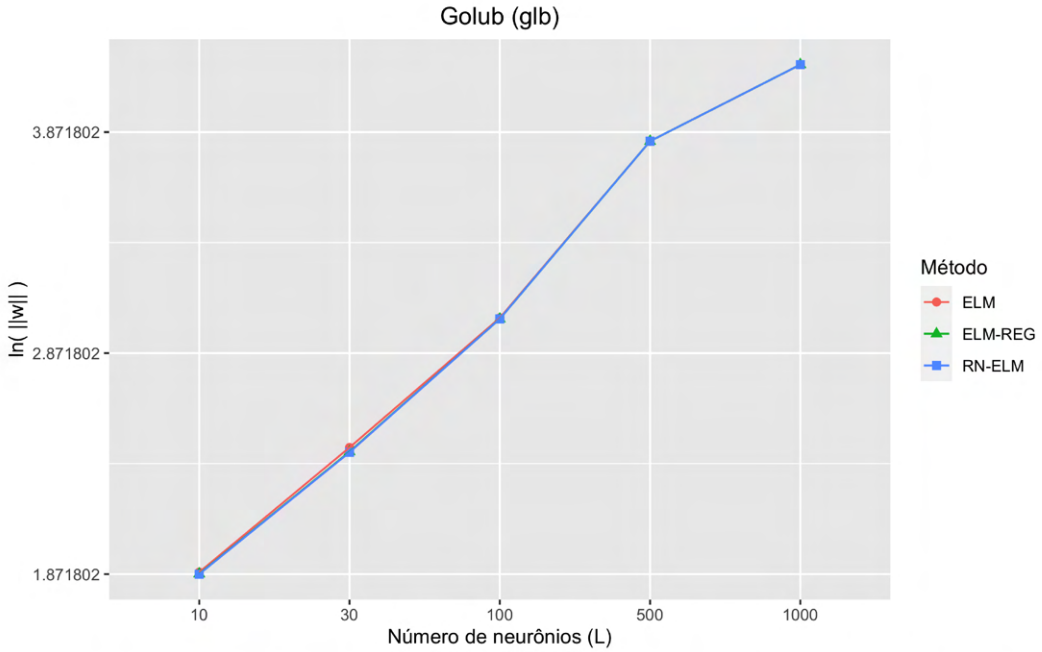


Figura 4.24: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

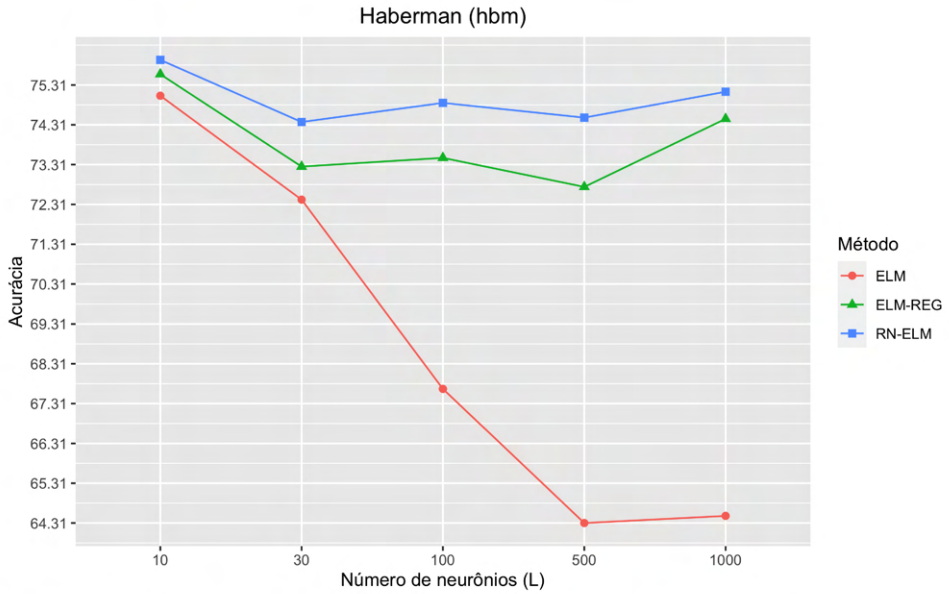


Figura 4.25: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

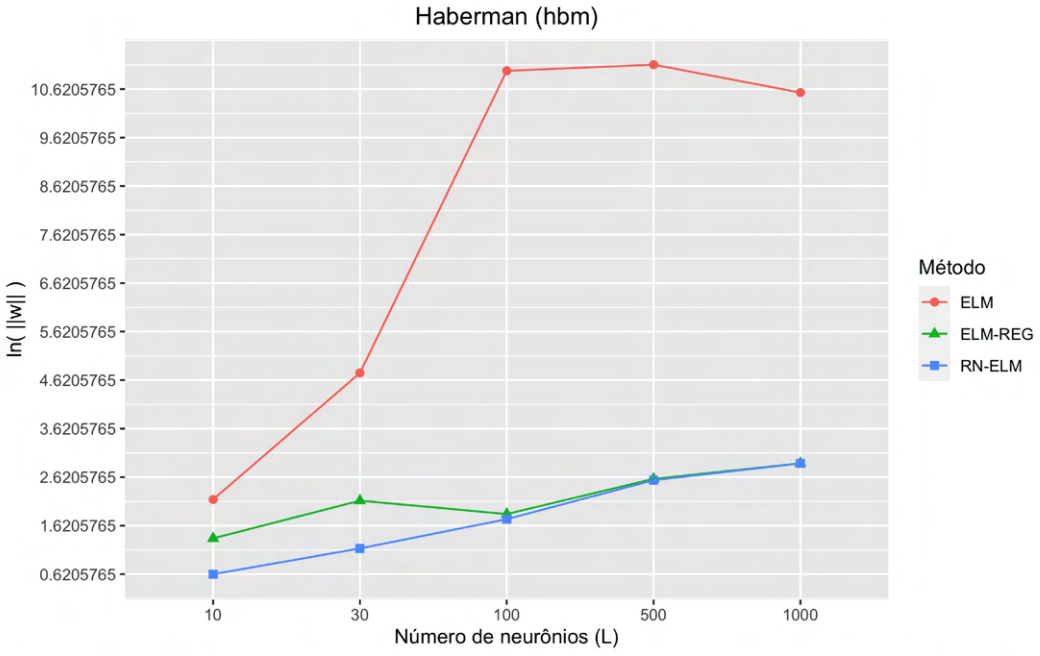


Figura 4.26: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

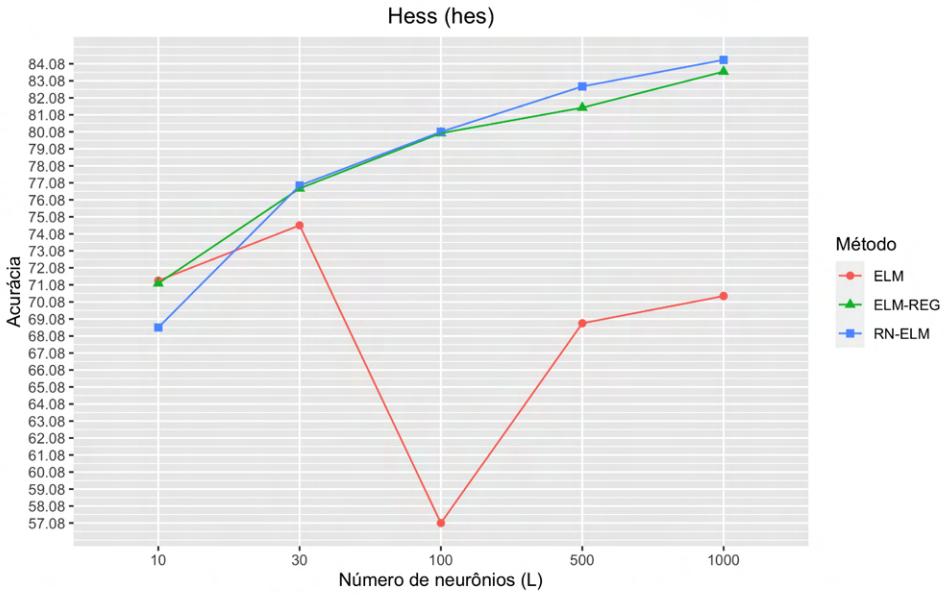


Figura 4.27: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

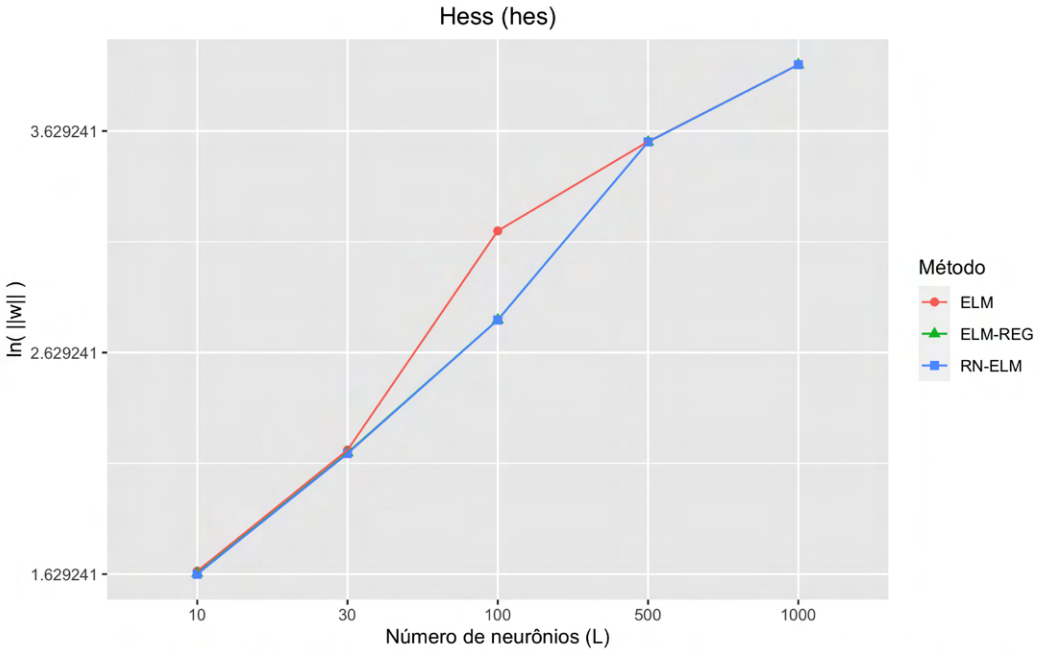


Figura 4.28: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

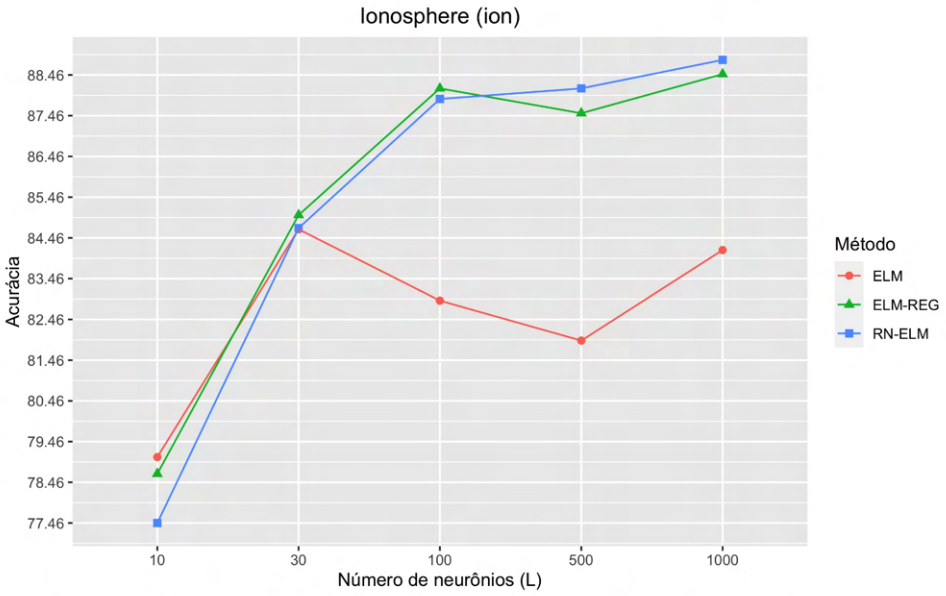


Figura 4.29: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

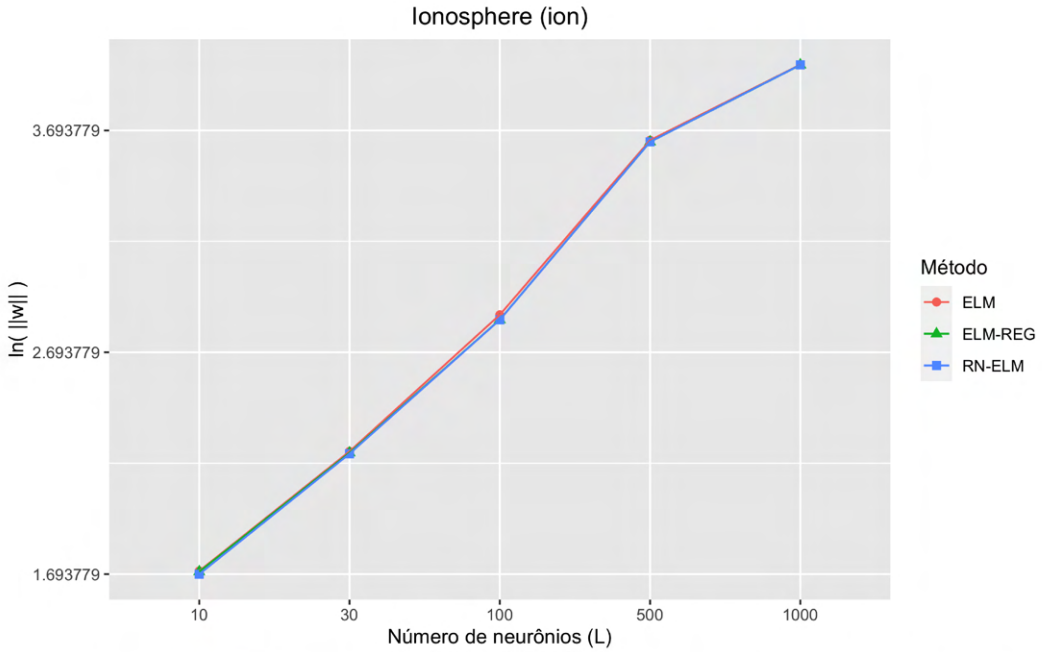


Figura 4.30: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

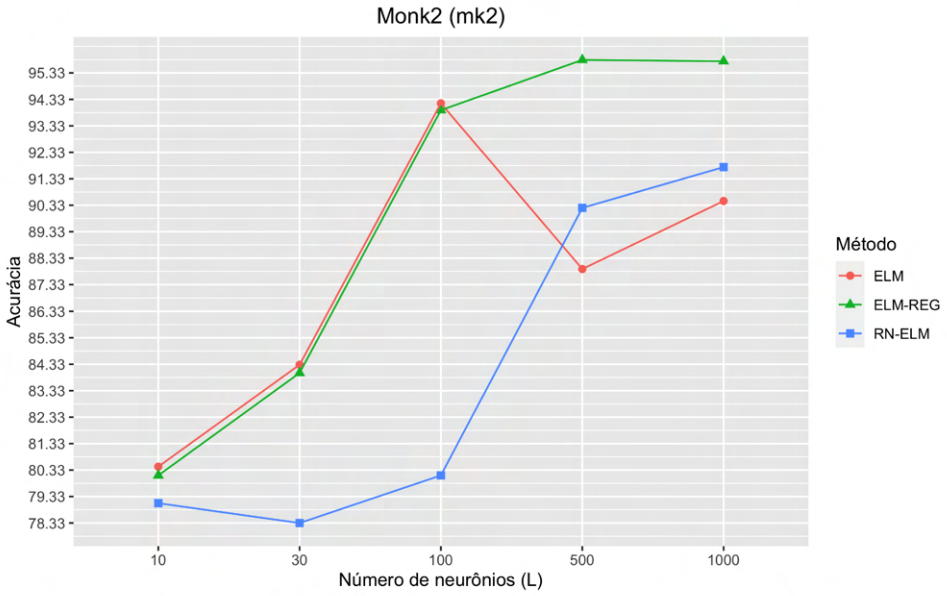


Figura 4.31: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

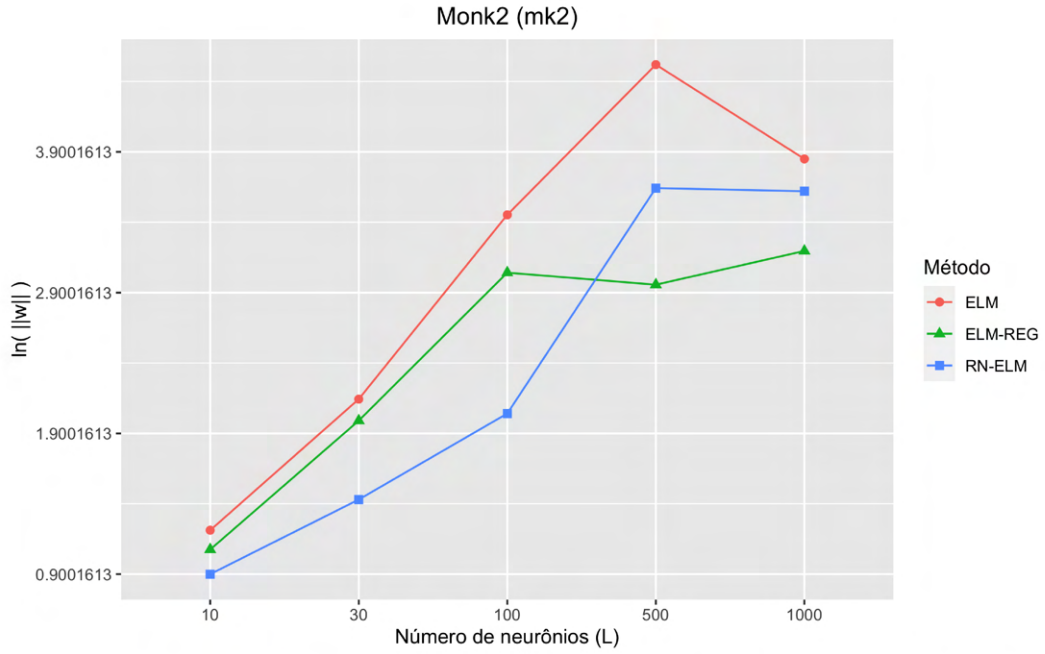


Figura 4.32: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

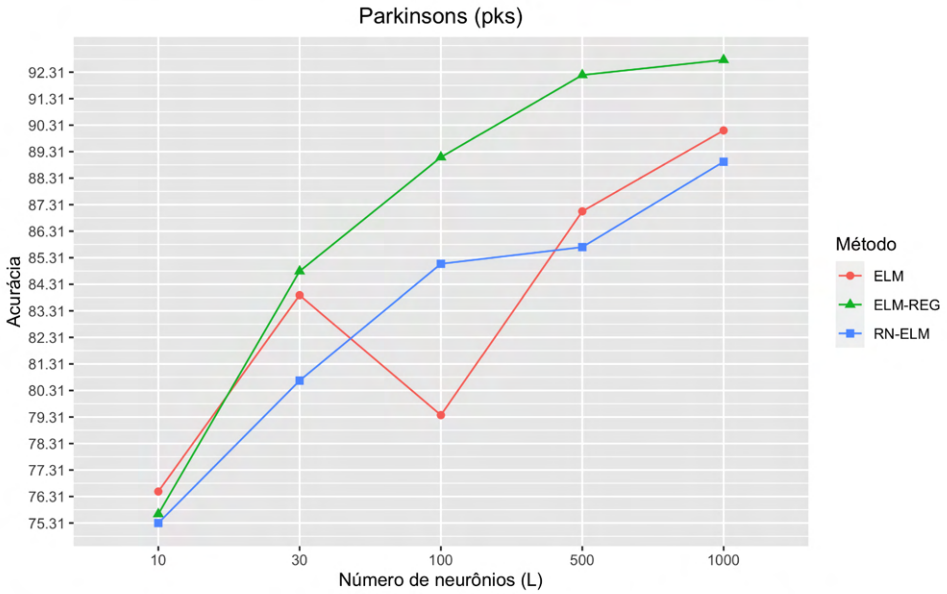


Figura 4.33: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

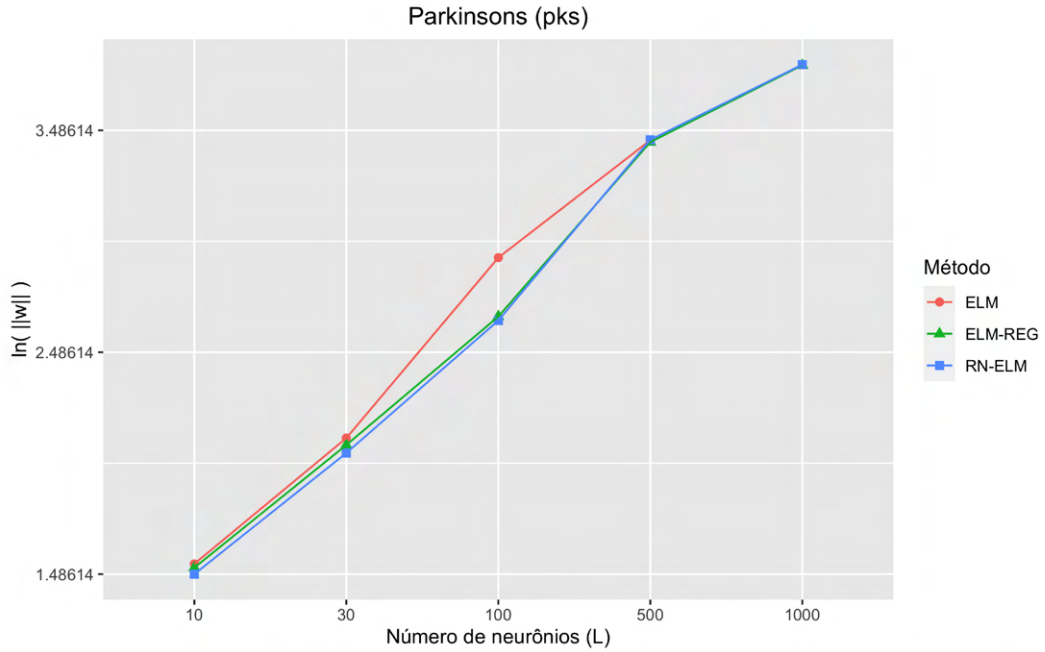


Figura 4.34: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

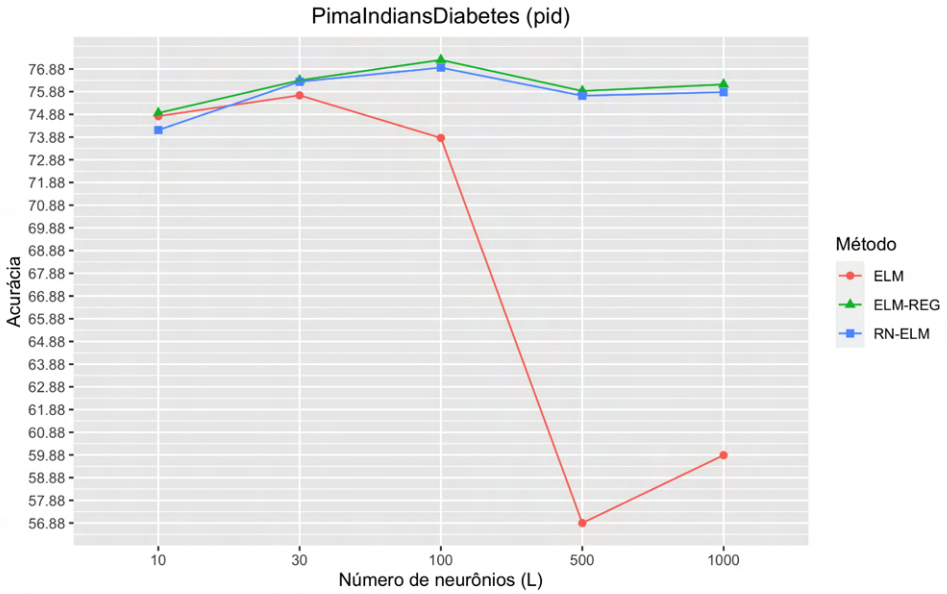


Figura 4.35: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

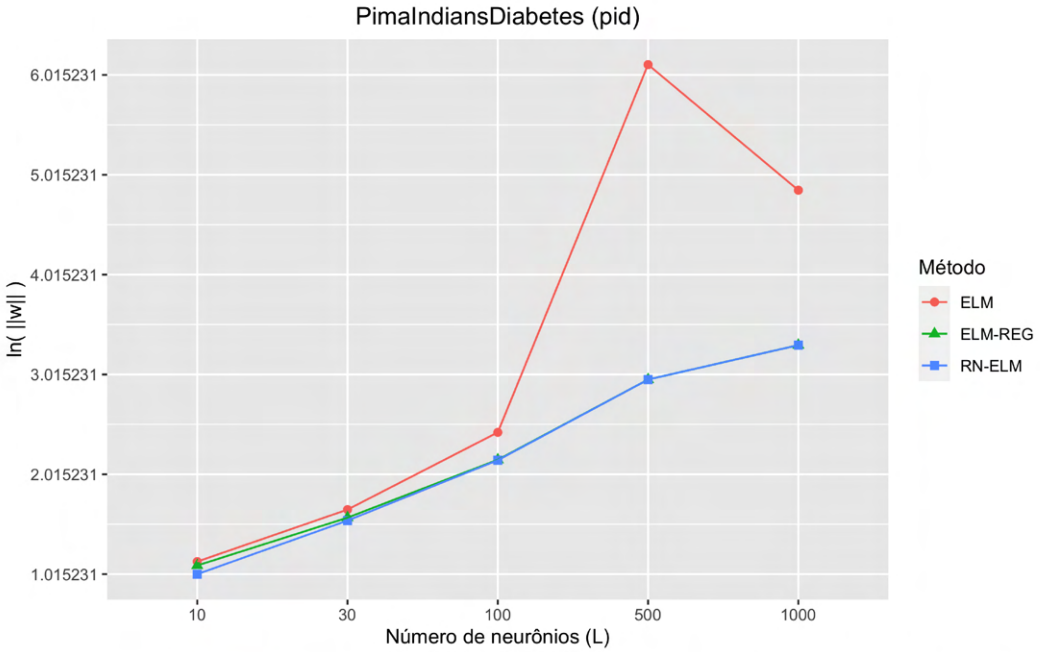


Figura 4.36: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

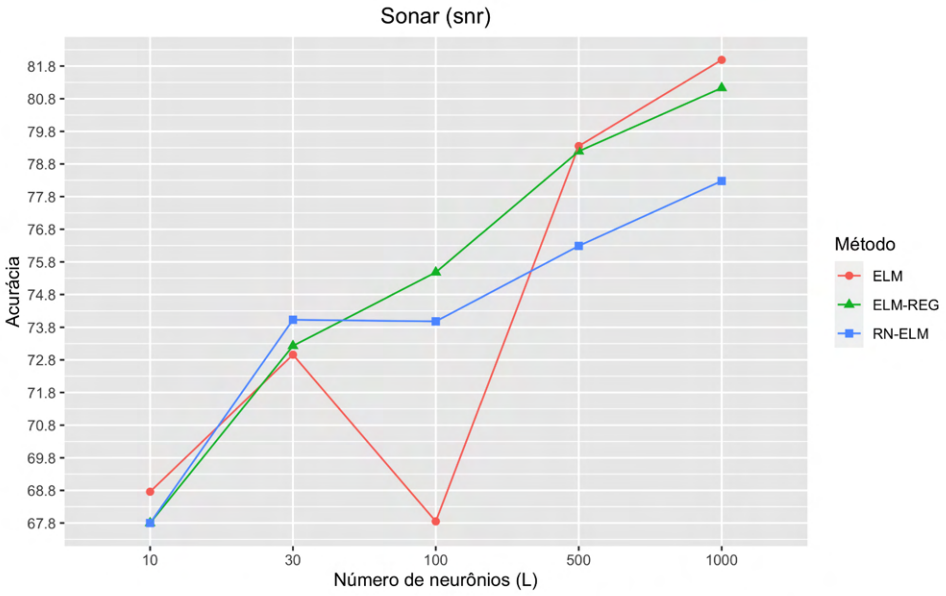


Figura 4.37: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

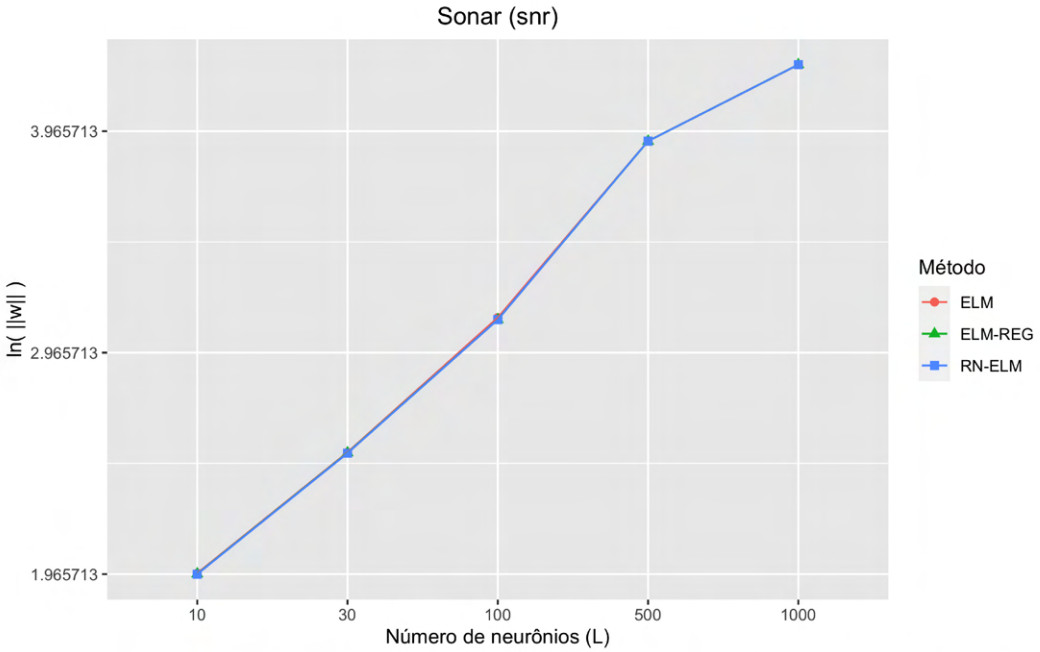


Figura 4.38: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

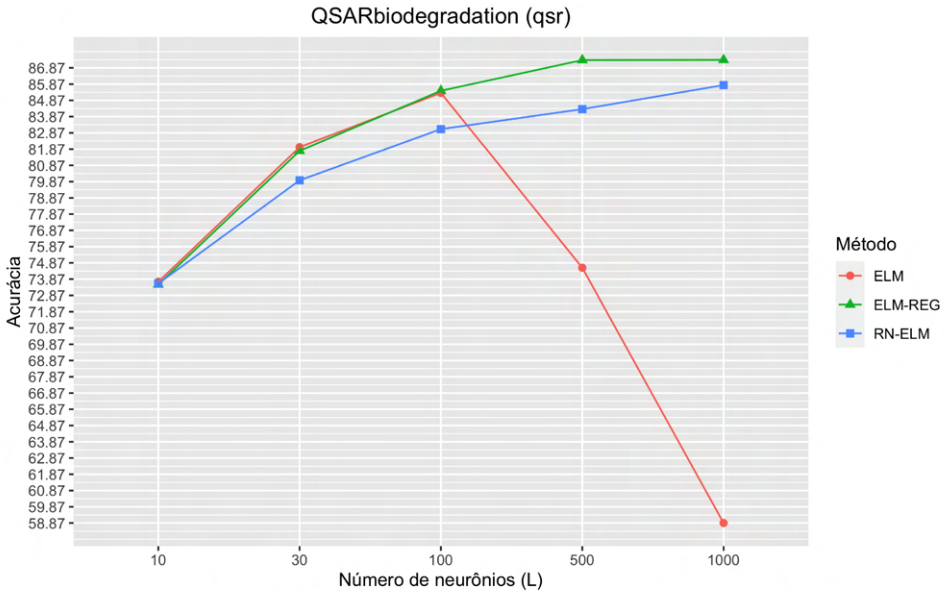


Figura 4.39: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

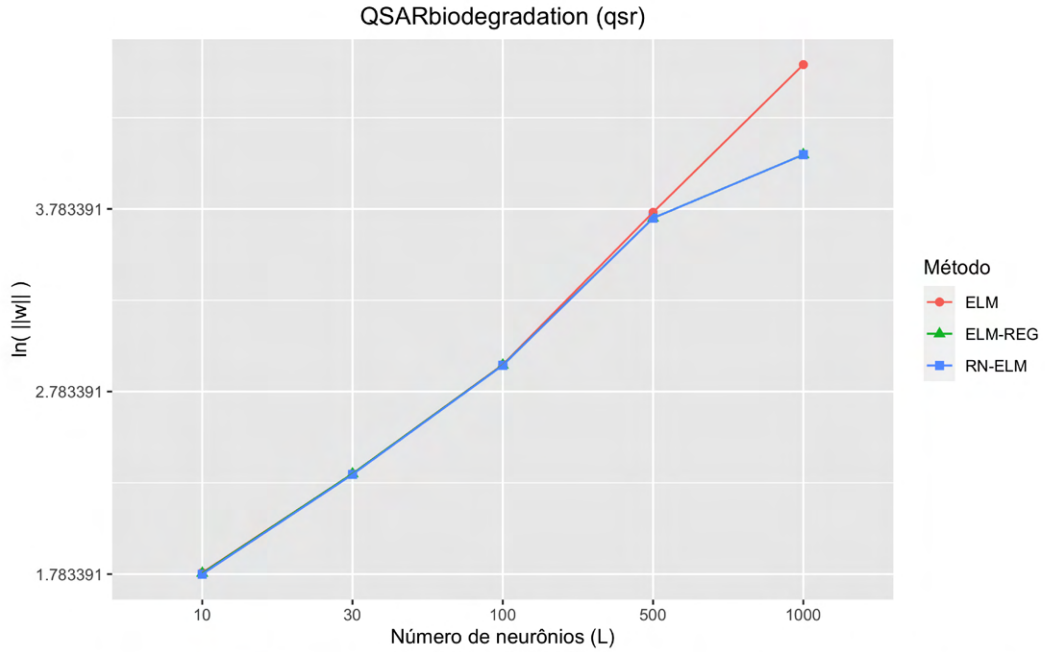


Figura 4.40: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

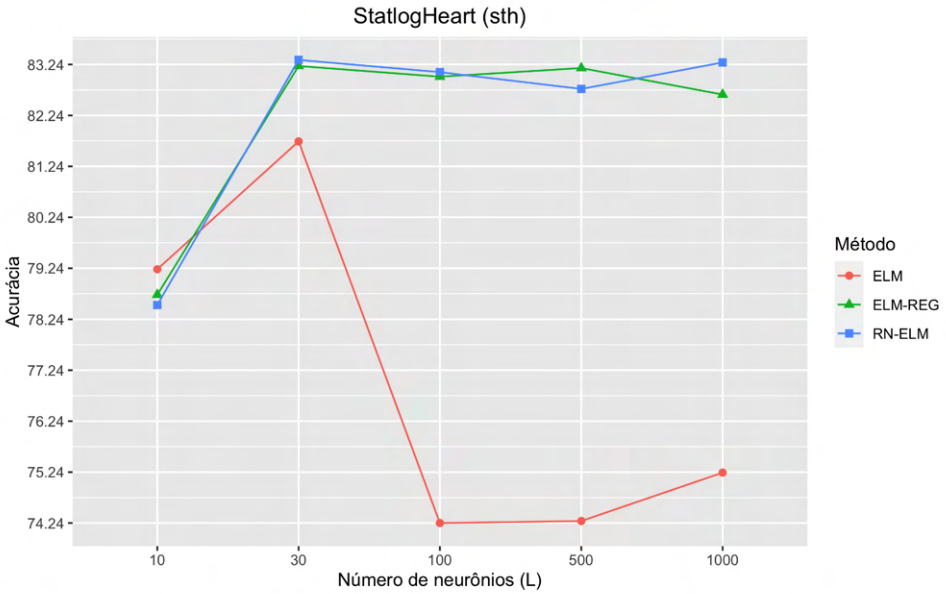


Figura 4.41: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

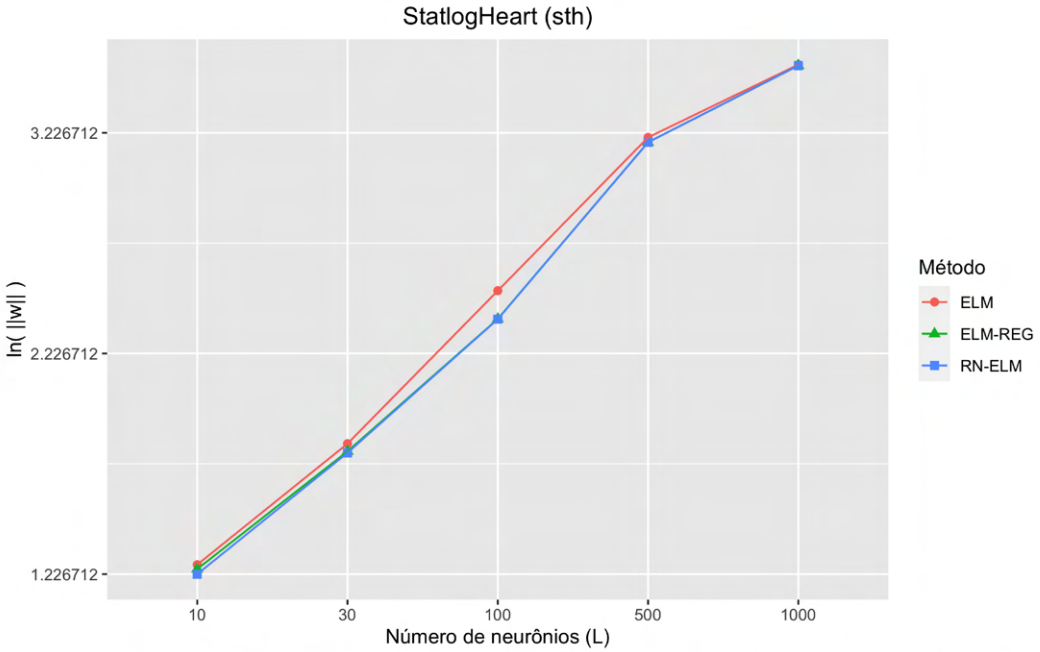


Figura 4.42: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

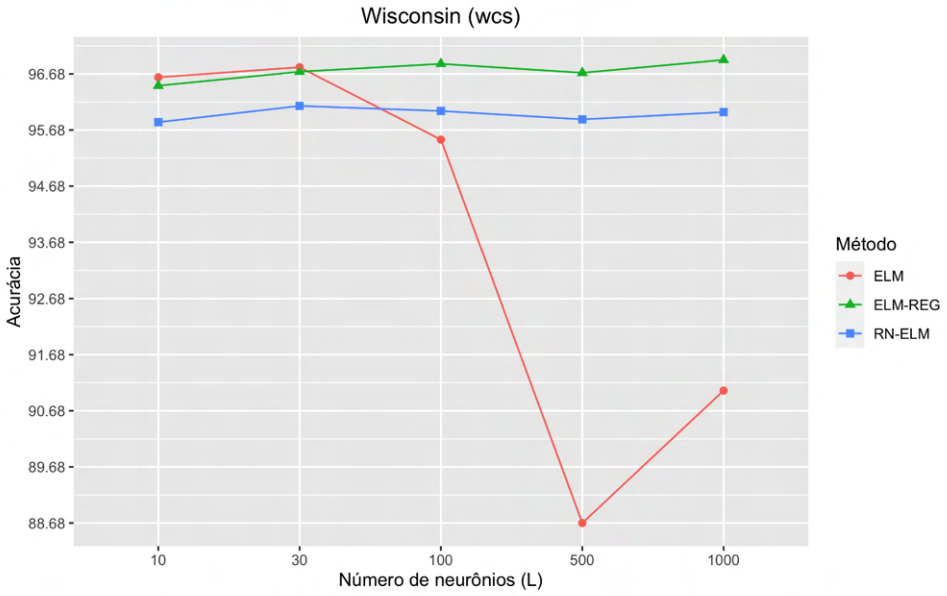


Figura 4.43: Resultado da acurácia média dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

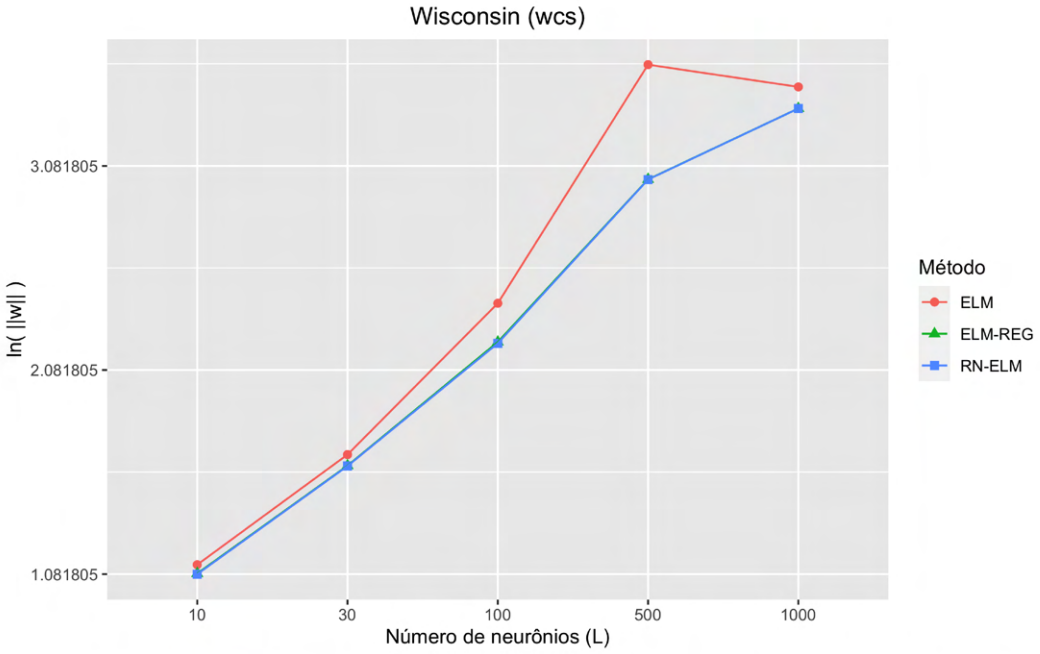


Figura 4.44: Resultado do logaritmo natural da norma dos pesos dos modelos ELM, ELM-REG e RN-ELM em relação ao número de neurônios (L) da RNA.

Tabela 4.2: Acurácia do conjunto de teste (média ± desvio padrão). Os resultados são apresentados de acordo com o modelo, número de neurônios na camada oculta (L) e a base de dados.

L	ELM	ELM-REG	RN-ELM	L	ELM	ELM-REG	RN-ELM
apd				**aud**			
10	87.08 ± 4.62	86.88 ± 5.94	85.10 ± 5.90	10	77.95 ± 3.05	77.73 ± 3.68	76.52 ± 3.21
30	79.90 ± 6.07	86.56 ± 5.51	83.65 ± 5.04	30	79.41 ± 2.30	80.36 ± 2.36	79.21 ± 3.08
100	56.46 ± 7.95	87.40 ± 5.53	83.23 ± 7.59	100	78.28 ± 2.78	81.25 ± 2.37	80.04 ± 2.34
500	62.81 ± 9.19	85.94 ± 5.67	75.62 ± 9.12	500	57.10 ± 3.49	81.55 ± 2.41	80.17 ± 2.14
1000	64.48 ± 10.33	86.98 ± 4.93	65.31 ± 10.77	1000	56.71 ± 3.48	80.78 ± 2.15	79.24 ± 1.92
aca				**bcr**			
10	82.91 ± 3.24	82.96 ± 3.13	82.37 ± 3.19	10	93.14 ± 2.25	92.87 ± 2.62	93.08 ± 2.52
30	85.97 ± 2.38	86.18 ± 2.46	86.18 ± 2.88	30	95.87 ± 1.57	96.04 ± 1.51	96.55 ± 1.28
100	84.94 ± 1.83	86.96 ± 1.99	86.84 ± 2.16	100	96.04 ± 1.53	96.69 ± 1.33	96.71 ± 1.17
500	51.76 ± 3.37	86.78 ± 2.54	86.73 ± 2.33	500	86.61 ± 2.42	97.43 ± 0.93	97.19 ± 1.05
1000	56.18 ± 4.78	86.92 ± 2.41	86.86 ± 1.98	1000	94.76 ± 1.59	97.68 ± 1.05	97.25 ± 1.05
bpa				**drp**			
10	67.98 ± 4.25	67.82 ± 4.40	66.19 ± 4.54	10	62.40 ± 2.94	62.30 ± 3.08	62.49 ± 3.29
30	69.17 ± 4.74	70.06 ± 4.78	68.81 ± 4.69	30	68.58 ± 1.90	68.71 ± 1.98	66.84 ± 2.18
100	64.23 ± 4.69	68.78 ± 4.47	68.78 ± 4.52	100	70.64 ± 2.33	70.77 ± 2.43	69.06 ± 2.53
500	54.04 ± 5.96	69.81 ± 4.46	70.83 ± 3.82	500	60.92 ± 2.40	72.20 ± 2.28	70.15 ± 2.64
1000	54.90 ± 5.57	70.26 ± 4.02	70.03 ± 4.80	1000	53.42 ± 2.93	72.60 ± 1.76	70.68 ± 2.05
ec1				**glb**			
10	88.05 ± 3.60	86.93 ± 3.60	86.53 ± 2.72	10	69.09 ± 10.57	72.42 ± 8.85	68.79 ± 10.46
30	90.63 ± 2.35	90.43 ± 2.17	88.55 ± 3.13	30	67.27 ± 10.09	74.55 ± 8.90	72.12 ± 9.53
100	86.93 ± 3.10	90.43 ± 2.51	89.67 ± 3.11	100	67.27 ± 11.47	77.27 ± 8.52	75.15 ± 9.45
500	69.11 ± 5.08	90.40 ± 2.58	89.11 ± 2.96	500	72.08 ± 9.40	75.97 ± 11.40	74.03 ± 8.74
1000	72.05 ± 5.40	89.67 ± 2.61	88.94 ± 2.86	1000	75.61 ± 7.50	78.48 ± 7.25	74.85 ± 8.75
hbm				**hes**			
10	75.04 ± 3.27	75.58 ± 3.62	75.94 ± 3.35	10	71.33 ± 6.32	71.17 ± 7.87	68.58 ± 8.30
30	72.43 ± 3.76	73.26 ± 4.28	74.38 ± 4.12	30	74.58 ± 6.98	76.75 ± 7.29	76.92 ± 6.81
100	67.68 ± 4.83	73.48 ± 4.34	74.86 ± 4.58	100	57.08 ± 7.25	80.00 ± 5.49	80.08 ± 5.63
500	64.31 ± 4.60	72.75 ± 4.21	74.49 ± 3.99	500	68.83 ± 7.30	81.50 ± 6.58	82.75 ± 6.27
1000	64.49 ± 3.35	74.46 ± 4.15	75.14 ± 4.16	1000	70.43 ± 6.45	83.62 ± 5.41	84.31 ± 4.91
ion				**mk2**			
10	79.08 ± 4.12	78.67 ± 4.34	77.46 ± 4.28	10	80.46 ± 3.02	80.13 ± 3.49	79.08 ± 3.23
30	84.67 ± 3.20	85.02 ± 3.33	84.70 ± 3.19	30	84.31 ± 4.05	84.00 ± 4.48	78.33 ± 3.29
100	82.92 ± 4.05	88.13 ± 3.15	87.87 ± 2.60	100	94.18 ± 2.48	93.92 ± 2.34	80.13 ± 3.99
500	81.94 ± 4.21	87.52 ± 2.77	88.13 ± 2.43	500	87.92 ± 4.29	95.82 ± 1.53	90.23 ± 3.75
1000	84.16 ± 2.56	88.48 ± 2.95	88.83 ± 2.69	1000	90.49 ± 2.48	95.77 ± 1.19	91.77 ± 3.04
pks				**pid**			
10	76.50 ± 6.43	75.65 ± 6.19	75.31 ± 5.90	10	74.80 ± 2.70	74.94 ± 2.88	74.19 ± 2.93
30	83.90 ± 5.30	84.80 ± 5.57	80.68 ± 5.39	30	75.72 ± 2.25	76.38 ± 2.26	76.32 ± 2.10
100	79.38 ± 6.54	89.10 ± 4.29	85.08 ± 4.57	100	73.84 ± 3.16	77.28 ± 2.79	76.94 ± 2.65
500	87.06 ± 4.23	92.20 ± 3.52	85.71 ± 4.07	500	56.88 ± 3.26	75.91 ± 2.56	75.70 ± 2.67
1000	90.11 ± 4.25	92.77 ± 3.27	88.93 ± 4.60	1000	59.87 ± 2.57	76.20 ± 2.10	75.86 ± 2.19
snr				**qsr**			
10	68.76 ± 7.87	67.80 ± 7.57	67.80 ± 8.13	10	73.71 ± 3.19	73.54 ± 3.33	73.60 ± 3.57
30	72.96 ± 4.80	73.23 ± 5.85	74.03 ± 5.89	30	81.99 ± 2.58	81.76 ± 2.75	79.95 ± 2.67
100	67.85 ± 5.59	75.48 ± 5.80	73.98 ± 4.44	100	85.33 ± 1.66	85.46 ± 1.62	83.10 ± 1.98
500	79.35 ± 3.43	79.19 ± 3.87	76.29 ± 4.43	500	74.57 ± 2.61	87.35 ± 1.62	84.33 ± 2.07
1000	81.99 ± 4.87	81.13 ± 5.48	78.28 ± 5.29	1000	58.87 ± 5.17	87.36 ± 1.42	85.81 ± 1.68
sth				**wcs**			
10	79.22 ± 4.10	78.72 ± 4.08	78.52 ± 4.51	10	96.62 ± 1.10	96.47 ± 1.17	95.82 ± 1.22
30	81.73 ± 4.68	83.21 ± 4.17	83.33 ± 3.99	30	96.80 ± 1.25	96.72 ± 1.20	96.11 ± 1.31
100	74.24 ± 5.37	83.00 ± 4.84	83.09 ± 4.46	100	95.51 ± 1.49	96.86 ± 1.09	96.02 ± 1.37
500	74.28 ± 3.64	83.17 ± 3.56	82.76 ± 4.08	500	88.68 ± 2.62	96.70 ± 1.04	95.87 ± 1.48
1000	75.23 ± 4.34	82.65 ± 4.17	83.28 ± 3.86	1000	91.04 ± 1.92	96.93 ± 1.13	96.00 ± 1.12

Tabela 4.3: Norma dos pesos por modelo (média ± desvio padrão). As normas são apresentadas de acordo com o modelo, número de neurônios na camada oculta (L) e a base de dados.

L	ELM	ELM-REG	RN-ELM	L	ELM	ELM-REG	RN-ELM
apd				**aud**			
10	5.71 ± 2.46	3.22 ± 0.96	2.65 ± 0.19	10	4.07 ± 0.16	4.01 ± 0.15	3.91 ± 0.13
30	28.23 ± 10.57	4.74 ± 0.73	4.53 ± 0.11	30	6.96 ± 0.15	6.77 ± 0.14	6.70 ± 0.12
100	913.81 ± 337.86	8.16 ± 0.15	8.18 ± 0.18	100	13.79 ± 0.42	12.25 ± 0.11	12.25 ± 0.11
500	127.27 ± 52.97	18.27 ± 0.10	37.57 ± 38.35	500	37505.15 ± 5535.61	27.37 ± 0.13	27.37 ± 0.13
1000	72.32 ± 29.68	25.82 ± 0.13	93.75 ± 44.66	1000	20508.92 ± 4048.61	38.73 ± 0.11	38.73 ± 0.10
aca				**bcr**			
10	3.69 ± 0.15	3.67 ± 0.16	3.53 ± 0.15	10	5.20 ± 0.11	5.18 ± 0.10	5.13 ± 0.10
30	6.25 ± 0.10	6.17 ± 0.11	6.12 ± 0.10	30	8.90 ± 0.11	8.87 ± 0.11	8.84 ± 0.11
100	11.50 ± 0.13	11.20 ± 0.11	11.19 ± 0.11	100	16.15 ± 0.13	16.07 ± 0.13	16.06 ± 0.13
500	447.38 ± 226.90	25.00 ± 0.13	24.99 ± 0.13	500	36.60 ± 0.18	35.92 ± 0.10	35.91 ± 0.10
1000	91.87 ± 37.98	35.27 ± 0.13	35.27 ± 0.13	1000	50.85 ± 0.14	50.77 ± 0.14	50.77 ± 0.14
bpa				**drp**			
10	3.65 ± 0.51	3.21 ± 0.50	2.41 ± 0.15	10	4.23 ± 0.11	4.21 ± 0.11	4.10 ± 0.11
30	8.61 ± 1.30	5.39 ± 1.29	4.19 ± 0.12	30	7.36 ± 0.13	7.27 ± 0.15	7.06 ± 0.13
100	59.13 ± 9.38	8.78 ± 1.86	7.64 ± 0.13	100	13.59 ± 0.17	13.05 ± 0.20	12.88 ± 0.13
500	884.34 ± 278.99	18.54 ± 5.03	17.09 ± 0.10	500	73.07 ± 5.33	28.91 ± 0.14	28.86 ± 0.13
1000	454.34 ± 153.56	24.20 ± 0.12	24.18 ± 0.10	1000	368.93 ± 31.32	40.80 ± 0.09	40.78 ± 0.10
ec1				**glb**			
10	4.00 ± 1.02	3.40 ± 0.66	2.75 ± 0.19	10	6.56 ± 0.11	6.51 ± 0.11	6.50 ± 0.11
30	9.74 ± 2.71	5.83 ± 1.34	4.53 ± 0.15	30	11.53 ± 0.20	11.31 ± 0.18	11.28 ± 0.15
100	93.38 ± 19.78	8.70 ± 1.18	8.26 ± 0.12	100	20.72 ± 0.13	20.63 ± 0.12	20.63 ± 0.12
500	1112.86 ± 260.47	18.40 ± 0.30	18.27 ± 0.12	500	46.10 ± 0.11	46.10 ± 0.11	46.10 ± 0.11
1000	513.86 ± 133.58	26.13 ± 0.68	25.86 ± 0.16	1000	65.18 ± 0.13	65.18 ± 0.13	65.18 ± 0.13
hbm				**hes**			
10	8.68 ± 3.62	3.89 ± 2.56	1.86 ± 0.12	10	5.17 ± 0.13	5.12 ± 0.13	5.10 ± 0.12
30	117.83 ± 46.32	8.47 ± 12.19	3.16 ± 0.12	30	8.93 ± 0.14	8.82 ± 0.14	8.79 ± 0.14
100	59692.60 ± 38473.75	6.42 ± 1.21	5.78 ± 0.14	100	24.02 ± 3.54	16.07 ± 0.15	16.06 ± 0.15
500	67824.21 ± 14660.07	13.26 ± 1.02	12.90 ± 0.12	500	35.94 ± 0.14	35.88 ± 0.13	35.88 ± 0.13
1000	38131.04 ± 6231.65	18.28 ± 0.14	18.26 ± 0.13	1000	50.86 ± 0.12	50.84 ± 0.12	50.84 ± 0.12
ion				**mk2**			
10	5.52 ± 0.13	5.50 ± 0.13	5.44 ± 0.12	10	3.36 ± 0.49	2.93 ± 0.42	2.46 ± 0.13
30	9.46 ± 0.14	9.40 ± 0.14	9.36 ± 0.13	30	8.53 ± 2.62	7.32 ± 2.36	4.18 ± 0.17
100	17.51 ± 0.12	17.13 ± 0.10	17.12 ± 0.09	100	31.60 ± 4.72	20.95 ± 8.19	7.70 ± 0.15
500	38.46 ± 0.17	38.19 ± 0.14	38.18 ± 0.14	500	91.83 ± 11.95	19.23 ± 3.07	38.20 ± 18.44
1000	54.08 ± 0.13	54.01 ± 0.13	54.01 ± 0.13	1000	47.00 ± 4.93	24.44 ± 0.64	37.36 ± 19.09
pks				**pid**			
10	4.63 ± 0.13	4.55 ± 0.14	4.42 ± 0.11	10	3.13 ± 0.16	3.01 ± 0.17	2.76 ± 0.13
30	8.16 ± 0.21	7.91 ± 0.27	7.63 ± 0.14	30	5.27 ± 0.25	4.86 ± 0.19	4.72 ± 0.11
100	18.40 ± 1.42	14.11 ± 0.30	13.86 ± 0.11	100	11.42 ± 0.58	8.68 ± 0.13	8.64 ± 0.13
500	31.23 ± 0.15	30.97 ± 0.15	31.29 ± 0.39	500	454.02 ± 57.82	19.38 ± 0.13	19.38 ± 0.13
1000	43.86 ± 0.15	43.79 ± 0.15	43.91 ± 0.17	1000	129.10 ± 9.32	27.36 ± 0.14	27.35 ± 0.14
qsr				**snr**			
10	6.00 ± 0.13	5.98 ± 0.13	5.95 ± 0.13	10	7.17 ± 0.13	7.15 ± 0.13	7.14 ± 0.13
30	10.32 ± 0.14	10.31 ± 0.14	10.27 ± 0.13	30	12.38 ± 0.11	12.35 ± 0.12	12.33 ± 0.11
100	18.74 ± 0.14	18.70 ± 0.14	18.67 ± 0.14	100	22.74 ± 0.13	22.54 ± 0.12	22.53 ± 0.12
500	43.15 ± 0.21	41.81 ± 0.13	41.80 ± 0.13	500	50.47 ± 0.13	50.46 ± 0.13	50.45 ± 0.13
1000	96.89 ± 28.06	59.18 ± 0.14	59.17 ± 0.14	1000	71.27 ± 0.13	71.27 ± 0.13	71.26 ± 0.13
sth				**wcs**			
10	3.56 ± 0.12	3.49 ± 0.15	3.41 ± 0.13	10	3.09 ± 0.13	2.96 ± 0.15	2.95 ± 0.14
30	6.16 ± 0.14	5.95 ± 0.12	5.91 ± 0.12	30	5.30 ± 0.15	5.02 ± 0.14	5.01 ± 0.15
100	12.32 ± 0.33	10.84 ± 0.14	10.83 ± 0.14	100	11.12 ± 0.47	9.20 ± 0.25	9.14 ± 0.11
500	24.68 ± 0.15	24.14 ± 0.12	24.14 ± 0.12	500	35.80 ± 4.47	20.41 ± 0.14	20.39 ± 0.13
1000	34.32 ± 0.12	34.18 ± 0.12	34.17 ± 0.12	1000	32.10 ± 0.74	28.88 ± 0.12	28.88 ± 0.12

Tabela 4.4: O número de hiperesferas e o número de amostras sintéticas em cada hiperesfera. Os dados do modelo RN-ELM são apresentados de acordo com número de neurônios na camada oculta (L) e a base de dados.

L	# hiperesferas (nH)	# amostras sintéticas (nA)	L	# hiperesferas	# amostras sintéticas
apd			**aud**		
10	27	40	10	249	4
30	30	36	30	262	32
100	30	12	100	265	16
500	29	40	500	268	12
1000	28	28	1000	266	20
aca			**bcr**		
10	478	4	10	168	20
30	593	36	30	212	40
100	585	20	100	225	40
500	561	20	500	224	20
1000	564	36	1000	229	40
bpa			**drp**		
10	285	20	10	959	8
30	309	32	30	1056	4
100	318	8	100	1099	16
500	326	4	500	1111	4
1000	319	28	1000	1110	4
ec1			**glb**		
10	74	8	10	46	12
30	78	4	30	69	36
100	75	16	100	83	36
500	74	20	500	92	36
1000	74	36	1000	93	32
hbm			**hes**		
10	135	36	10	77	32
30	138	8	30	108	28
100	135	40	100	139	40
500	137	16	500	144	36
1000	137	40	1000	142	24
ion			**mk2**		
10	143	4	10	272	12
30	170	4	30	198	8
100	177	40	100	150	4
500	180	32	500	149	8
1000	176	32	1000	150	12
pks			**pid**		
10	67	4	10	593	16
30	69	40	30	682	12
100	70	28	100	710	4
500	71	4	500	712	12
1000	73	36	1000	719	12
qsr			**snr**		
10	588	16	10	211	40
30	684	4	30	451	4
100	693	4	100	705	8
500	693	4	500	907	8
1000	691	4	1000	925	4
sth			**wcs**		
10	219	8	10	94	16
30	300	36	30	106	4
100	350	28	100	112	4
500	356	4	500	117	20
1000	352	32	1000	115	28

4.2.2.2 Teste de Friedman

A comparação direta do desempenho entre os classificadores pode ser enviesada por causa do desbalanceamento das amostras de cada classe e também pelo desempenho distinto de cada base de dados. Assim, o resultado não é suficiente para uma conclusão confiável na comparação dos classificadores (Stapor, 2017). Porém, é proposto por Japkowicz & Shah (2011) e Demšar (2006) um conjunto de testes estatísticos para comparação de classificadores. O teste de Friedman (1937, 1940) é um teste não paramétrico para comparação de vários classificadores para várias bases de dados, em que cada algoritmo é ranqueado para cada base de dados. O algoritmo de melhor desempenho recebe o *rank* 1, o segundo melhor o *rank* 2 e assim, sucessivamente, para todos os outros algoritmos. E em caso de empate é utilizado o *rank* médio. Para o classificador j, é somado o *rank* de todas as bases de dados. O resultado estatístico do teste de Friedman é expresso por χ_F^2 na Equação 4.2.

$$\chi_F^2 = \left[\frac{12}{n * k * (k+1)} * \sum_{j=1}^{k} (R_{.j})^2 \right] - 3 * n * (k+1) \tag{4.2}$$

onde n é o número de bases de dados, k o número de classificadores e $R_{.j}$ é a soma dos *ranks* do classificador j para todas bases de dados. O valor obtido de χ_F^2 é comparado em uma tabela disponível em (Japkowicz & Shah, 2011) de acordo com o nível de significância. Ao comparar o valor de χ_F^2 com o valor na tabela e este for maior, rejeita-se a hipótese nula (H_0) de equivalência entre os classificadores.

4.2.2.3 Resultado do teste de Friedman

Neste trabalho, para obter o p-valor do teste de Friedman foi utilizado o pacote *stats* da linguagem R (R Core Team, 2019). Se o p-valor for menor que o nível de significância de $0,05$, então rejeita-se a hipótese nula (H_0) de igualdade entre os modelos de classificadores. O resultado do teste de Friedman para os modelos de treinamento ELM, ELM-REG e RN-ELM para diferentes configurações do número de neurônios na camada oculta (L) pode ser visto na Tabela 4.5, em que a hipótese de igualdade entre os modelos (H_0) pode ser rejeitada ao nível de significância de $0,05$ para $L = \{10, 30, 100, 500 \text{ e } 1000\}$.

O resultado do teste de Friedman aplicado na acurácia média da Tabela 4.2 pelo p-valor indica a existência de diferença significativa entre os modelos de treinamento de ELM, mas não é suficiente para descrever se os três modelos são diferentes ou apenas um deles tem desempenho diferente e os demais não há diferença significativa.

Tabela 4.5: Resultado do teste de Friedman (p-value) para diferentes números de neurônios na camada oculta (L).

L	***p*-value**
10	0.0004144
30	0.03974
100	4.604e-05
500	1.58e-05
1000	3.843e-05

O teste *post hoc* recomendado por Demšar (2006); Japkowicz & Shah (2011) é o teste de Nemenyi (Nemenyi, 1963) que compara cada classificador contra todos os outros e utiliza como referência a Diferença Crítica (*Critical Difference* (CD)).

4.2.2.4 Teste de Nemenyi

O teste de Nemenyi calcula o *rank* médio de cada classificador e obtém suas diferenças. CD é representada pela diferença do *rank*. A partir do *rank* médio de cada classificador é obtida sua diferença. Para o caso em que a diferença do rank médio é maior ou igual a CD, nós podemos dizer com apropriado nível de certeza que o desempenho dos dois classificadores correspondentes são significativamente diferentes um do outro.

Neste trabalho, para realizar o teste de Nemenyi foi utilizado o pacote PMCMR (*The Pairwise Multiple Comparison of Mean Ranks Package*) (Pohlert, 2014) da linguagem R que realiza a comparação múltipla. O Diagrama de Diferença Crítica (*Critical Difference Diagram* (CDD)) é apresentado em (Demšar, 2006) e permite a representação gráfica dos resultados. O CDD consiste em um eixo horizontal que representa o *rank* médio do classificador, em que o classificador mais a esquerda com menor valor representa o melhor resultado. O segmento acima do eixo representa a CD. Os modelos em que não existem diferenças significativas são conectados por uma linha horizontal.

4.2.2.5 Resultado do teste de Nemenyi

Os resultados do teste *post-hoc* de Nemenyi são apresentados nas Figuras 4.45, 4.46, 4.47, 4.48 e 4.49 em que é observado que para todos os casos avaliados, o modelo RN-ELM e ELM-REG não apresentam diferenças significativas entre eles. Ainda que o modelo ELM-REG tenha o melhor *rank* a distância do *rank* médio entre eles é menor que a CD. Para redes neurais com 100 ou mais neurônios na camada oculta foi observado que os modelos com regularização se distingue do modelo ELM padrão.

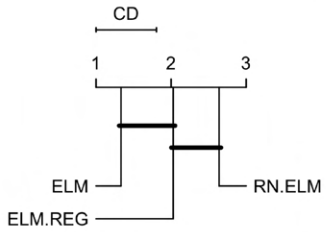


Figura 4.45: Diagrama de Diferença Crítica do modelo com 10 neurônios na camada oculta.

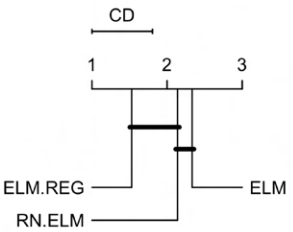


Figura 4.46: Diagrama de Diferença Crítica do modelo com 30 neurônios na camada oculta.

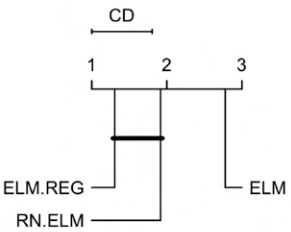


Figura 4.47: Diagrama de Diferença Crítica do modelo com 100 neurônios na camada oculta.

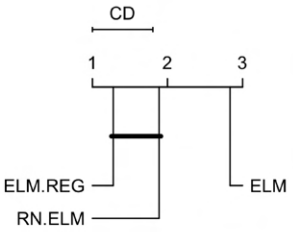


Figura 4.48: Diagrama de Diferença Crítica do modelo com 500 neurônios na camada oculta.

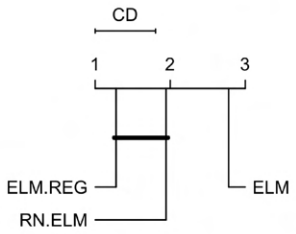


Figura 4.49: Diagrama de Diferença Crítica do modelo com 1000 neurônios na camada oculta.

4.2.3 Conclusão do capítulo

Para os experimentos com as bases sintéticas foi possível observar a redução da complexidade da rede neural utilizando o modelo proposto neste trabalho, RN-ELM, em relação ao modelo ELM padrão. Houve suavização da superfície de separação das bases de dados sintéticas com a utilização dos modelos com regularização.

Para os experimentos com bases reais foi observado que existem diferenças entre os modelos com regularização para o modelo ELM padrão. A diferença se destaca para as redes neurais com maior número de neurônios na camada oculta. Pelas análises estatísticas utilizadas o modelo RN-ELM não apresenta diferença significativa de acurácia em relação do modelo ELM-REG. Desta forma, reforça a capacidade de regularização por reamostragem do modelo proposto neste trabalho. Os modelos de regularização se mostram mais adequados para o ajuste da rede neural a complexidade do problema quando está rede neural tem grande número de neurônios na camada oculta, super dimensionada.

A definição do número de amostras sintéticas ruidosas geradas em cada hiperesfera mostrou-se adequado para as 18 bases de dados, visto que durante o treinamento o conjunto de amostras sintéticas variam de 4 até 40, em que é escolhida as amostras que geram melhor acurácia.

Capítulo 5

Conclusões e trabalhos futuros

5.1 Conclusões

Foi demonstrado formalmente neste trabalho que o treinamento de uma rede ELM com reamostragem local possui o mesmo efeito da regularização de Tikhonov. Este resultado segue o desenvolvimento apresentado por Bishop (1995), entretanto, a abordagem apresentada nesta tese com a reamostragem local com grafo leva a função de separação para a região de margem, sem a necessidade de cobrir exaustivamente todo o espaço de entrada. Os resultados apresentados neste trabalho mostram que tal abordagem reduz a norma dos pesos, indicando que os métodos geram soluções mais suaves, reduzindo os efeitos de *overfitting*. As métricas de desempenho também indicam que os resultados são estatisticamente equivalentes aos obtidos pelo ELM-REG com parâmetros de regularização obtidos com validação cruzada. O modelo apresentado neste trabalho pode ser considerado como um classificador de margem larga por utilizar os vetores de borda do grafo de Gabriel, que são semelhantes aos vetores de suporte da SVM (Torres, 2016).

A reamostragem orientada pela abordagem com grafo pode reduzir o custo de exploração do espaço de entrada em problemas com alta dimensão e elevado número de amostras. Embora o grafo precise ser gerado para localizar a reamostragem, ele é baseado em informações de pares que podem ser totalmente paralelizadas. A definição dos parâmetros do modelo é realizada a partir das informações do conjunto de treinamento e não precisa ser realizada iterativamente.

Nos experimentos realizados com bases de dados sintéticas na rede ELM, com 500 neurônios na camada escondida, foi possível visualizar o efeito de regularização pela suavização das superfícies de separação entre as classes do método proposto RN-ELM em relação ao padrão ELM. Já nos experimentos com as 18 bases de dados reais com diferentes números de amostras e características foram comparados os desempenhos dos modelos ELM padrão, ELM-REG e RN-ELM. As métricas utilizadas na comparação foram a acurácia e a norma dos pesos. O modelo ELM padrão tem redução do desempenho pela acurácia e aumento da norma dos pesos à medida que ocorre o aumento da complexidade, aumento do número de neurônios. Para os modelos que utilizam técnicas de regularização o efeito foi inverso, ou seja, aumento da acurácia e redução da norma dos pesos. Desta forma estes modelos são os mais aptos a reconhecer as características das bases de dados. O modelo proposto pode ser comparado a técnica de *data augmentation*, pois um conjunto de amostras sintéticas são adicionadas durante o treinamento no espaço de características e tem como referência dados de entrada com uma melhor representatividade.

O aumento do tempo de treinamento do modelo proposto pode ser identificado em operações de cálculo e tarefas computacionais. O cálculo da matriz pseudo-inversa utiliza amostras no espaço de características com alta dimensionalidade, levando a um maior custo computacional. A busca e avaliação de outros métodos que possam ser mais eficientes para o cálculo da pseudo-inversa pode melhorar o desempenho do método. Durante o treinamento da rede neural, o grafo é gerado sempre que as amostras do conjunto de treino são alteradas e isto contribui para o aumento do tempo de treinamento. Um estudo é necessário para avaliar sobre a atualização do grafo de Gabriel considerando apenas os vértices que foram inseridos e os removidos. Desta forma, o grafo pode ser atualizado ao invés de recriado o que reduzirá o tempo de obtenção do grafo e a demanda computacional. Pode ocorrer sobreposição das amostras sintéticas geradas em hiperesferas vizinhas que possuem região em comum o que aumenta o número de amostras geradas. É necessário avaliar se ocorre sobreposição entre as hiperesferas para evitar sobreposição de amostras. O aumento do tempo de treinamento do modelo proposto pode ser tratado, visto que as tarefas realizadas podem ser tratadas computacionalmente com paralelização e pelo desenvolvimento de novas técnicas.

5.2 Trabalhos futuros

Algumas sugestões para continuidade deste trabalho:

- Pesquisar por métodos que tenham melhor desempenho para o cálculo da matriz pseudo-inversa.
- Avaliar e desenvolver procedimentos para atualização do grafo de Gabriel a partir dos vértices adicionados e removidos. Assim não será necessário construir todo o grafo, mas apenas um reconstrução local.
- Desenvolver procedimentos para evitar a sobreposição de amostras sintéticas geradas entre hiperplanos vizinhos que possuam área de cobertura comum.
- Paralelizar a geração das amostras sintéticas nos hiperplanos.
- Aplicar o método proposto em redes neurais profundas.
- Alterar o método proposto para tratar de problemas de classificação multiclasses.

Referências

Abu-Mostafa, Y. (1989). Information theory, complexity and neural networks. *IEEE Communications Magazine*, **27**, 25–28. 55

Alcala-Fdez, J., Fernández, A., Luengo, J., Derrac, J. & García, S. (2011). Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Multiple-Valued Logic and Soft Computing*, **17**, 255–287. 51, 54

An, G. (1996). The effects of adding noise during backpropagation training on a generalization performance. *Neural Computation*, **8**, 643–674. 24

Araujo, L.R.G., Torres, L.C.B., Silvestre, L.J., Takahashi, C. & Braga, A.P. (2019). Regularization of extreme learning machines with information of spatial relations of the projected data. In *2019 6th International Conference on Control, Decision and Information Technologies (CoDIT)*, 593–597. 35

Barata, J.C.A. & Hussein, M.S. (2012). The moore–penrose pseudoinverse: A tutorial review of the theory. *Brazilian Journal of Physics*, **42**, 146–165. 36

Bazan, F.S.V. (2009). MÃ©todos para problemas inversos de grande porte. In S.B. de MatemÃ¡tica Aplicada e Computacional, ed., *Notas em MatemÃ¡tica Aplicada*. 30

Bennett, K.P. & Bredensteiner, E.J. (2000). Duality and geometry in svm classifiers. In *Proceedings of the Seventeenth International Conference on Machine Learning*, ICML '00, 57–64, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA. 20

Bishop, C.M. (1995a). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA. 18, 29, 44

Bishop, C.M. (1995b). *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA. 31

Bishop, C.M. (1995c). Regularization and Complexity Control in Feed-forward Networks. *Proceedings International Conference on Artificial Neural Networks ICANN'95*, **1**, 141–148. 40

Bishop, C.M. (1995). Training with noise is equivalent to tikhonov regularization. *Neural Computation*, **7**, 108–116. 20, 21, 24, 28, 31, 32, 41, 82

Borges, A.J. & Bazan, F.S.V. (2009). Um estudo numÃ©rico sobre tÃ©cnicas de regularizaÃ§Ã£o diretas e iterativas. In *XXXII Congresso Nacional de MatemÃ¡tica Aplicada e Computacional*. 30

Bouthillier, X., Konda, K., Vincent, P. & Memisevic, R. (2016). Dropout as data augmentation. 20

Braga, J. (2001). Numerical comparison between tikhonov regularization and singular value decomposition methods using the l curve criterion. *Journal of Mathematical Chemistry*, **29**, 151–161. 30

Calvetti, D., Morigi, S., Reichel, L. & Sgallari, F. (2000). Tikhonov regularization and the l-curve for large discrete ill-posed problems. *Journal of computational and applied mathematics*, **123**, 423–446. 30

Cao, M. & Qiao, P. (2008). Neural network committee-based sensitivity analysis strategy for geotechnical engineering problems. *Neural Computing and Applications*, **17**, 509–519. 38

Chandra, P. & Singh, Y. (2003). Regularization and feedforward artificial neural network training with noise. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, vol. 3, 2366–2371 vol.3. 21

Cover, T.M. (1965). Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Transactions on Electronic Computers*, **EC-14**, 326–334. 50

de Campos Velho, H.F. (2001). Problemas inversos: conceitos básicos e aplicações. *Anais do Encontro de Modelagem Computacional. Mini-curso*, 63–79. 31

DeGroot, M. & Schervish, M. (2012). *Probability and Statistics*. Addison-Wesley. 46

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *J. Mach. Learn. Res.*, **7**, 1–30. 79, 80

Deng, W., Zheng, Q. & Chen, L. (2009). Regularized extreme learning machine. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*, 389–395. 35, 36

DeVries, T. & Taylor, G.W. (2017). Dataset augmentation in feature space. 20

Dheeru, D. & Karra Taniskidou, E. (2017). Uci machine learning repository. 51, 54

Ding, S., Xu, X. & Nie, R. (2014). Extreme learning machine and its applications. *Neural Computing and Applications*, **25**, 549–556. 35

Duda, R.O., Hart, P.E. & Stork, D.G. (2000). *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2nd edn. 18, 29

Evans, M.J. & Rosenthal, J.S. (2004). *Probability and statistics: The science of uncertainty*. Macmillan. 38

Ferrari, S. & Stengel, R. (2005). Smooth function approximation using neural networks. *IEEE Transactions on Neural Networks*, **16**, 24–38. 36

Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. *Journal of the American Statistical Association*, **32**, 675–701. 56, 79

Friedman, M. (1940). A comparison of alternative tests of significance for the problem of m rankings. *The Annals of Mathematical Statistics*, **11**, 86–92. 56, 79

Gabriel, K. Ruben and Sokal, R.R. (1969). A New Statistical Approach to Geographic Variation Analysis. *Systematic Biology*, **18**, 259–278. 32

Garcia, L.P., de Carvalho, A.C. & Lorena, A.C. (2015). Effect of label noise in the complexity of classification problems. *Neurocomputing*, **160**, 108–119. 24, 28

Garcia, L.P., Lehmann, J., de Carvalho, A.C. & Lorena, A.C. (2019). New label noise injection methods for the evaluation of noise filters. *Knowledge-Based Systems*, **163**, 693–704. 24, 28

Geman, S., Bienenstock, E. & Doursat, R. (1992). Neural Networks and the Bias/Variance Dilemma. 21, 40

Geurts, P. (2010). *Bias vs Variance Decomposition for Regression and Classification*, 733–746. Springer US, Boston, MA. 40

Golub, G.H. & Von Matt, U. (1997). *Tikhonov regularization for large scale problems*. Citeseer. 30

Golub, G.H., Hansen, P.C. & O'Leary, D.P. (1999a). Tikhonov regularization and total least squares. *SIAM journal on matrix analysis and applications*, **21**, 185–194. 30

Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A. *et al.* (1999b). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, **286**, 531–537. 54

Hagiwara, K. (2002). Regularization learning, early stopping and biased estimator. *Neurocomputing*, **48**, 937–955. 20, 40

Haykin, S. (2009). *Neural Networks and Learning Machines*. Prentice Hall, 3rd edn. 18, 29, 30

Hess, K.R., Anderson, K., Symmans, W.F., Valero, V., Ibrahim, N., Mejia, J.A., Booser, D., Theriault, R.L., Buzdar, A.U., Dempsey, P.J. *et al.* (2006). Pharmacogenomic predictor of sensitivity to preoperative chemotherapy with paclitaxel and fluorouracil, doxorubicin, and cyclophosphamide in breast cancer. *Journal of clinical oncology*, **24**, 4236–4244. 54

Ho, K.I.J., Leung, A.C.S. & Sum, J. (2008). On weight-noise-injection training. In *ICONIP*. 27

Holmstrom, L. & Koistinen, P. (1992). Using additive noise in back-propagation training. *Trans. Neur. Netw.*, **3**, 24–38. 20, 24, 26

Huang, G.B., Zhu, Q.Y. & Siew, C.K. (2004). Extreme learning machine: a new learning scheme of feedforward neural networks. In *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, vol. 2, 985–990 vol.2. 35

Huang, G.B., Zhu, Q.Y. & Siew, C.K. (2006). Extreme learning machine: Theory and applications. *Neurocomputing*, **70**, 489 – 501. 35, 36, 49

Huang, G.B., Zhou, H., Ding, X. & Zhang, R. (2012). Extreme learning machine for regression and multiclass classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, **42**, 1–17. 37, 49, 50, 55

Inaba, F., Salles, E., Perron, S. & Caporossi, G. (2017). Dgr-elm-distributed generalized regularized elm for classification. *Neurocomputing*, **275**. 35

Isaev, I. & Dolenko, S. (2018). Training with noise addition in neural network solution of inverse problems: Procedures for selection of the optimal network. *Procedia Computer Science*, **123**, 171 – 176. 24

Japkowicz, N. & Shah, M. (2011). *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, USA. 79

Karsoliya, S. & Azad, M. (2012). Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. In *International Journal of Engineering Trends and Technology*. 55

Krogh, A. & Hertz, J.A. (1992). A simple weight decay can improve generalization. In J.E. Moody, S.J. Hanson & R.P. Lippmann, eds., *Advances in Neural Information Processing Systems 4*, 950–957, Morgan-Kaufmann. 20, 30

Kruglov, V. (2011). A generalization of weak law of large numbers. *Stochastic Analysis and Applications*, **29**, 674–683. 39

Lemley, J., Bazrafkan, S. & Corcoran, P. (2017). Smart augmentation learning an optimal data augmentation strategy. *Ieee Access*, **5**, 5858–5869. 20

Ludermir, T.B., Yamazaki, A. & Zanchettin, C. (2006). An optimization methodology for neural network weights and architectures. *IEEE Transactions on Neural Networks*, **17**, 1452–1459. 40

Matsuoka, K. (1992). Noise Injection into Inputs in Back-Propagation Learning. *IEEE Transactions on Systems, Man and Cybernetics*, **22**, 436–440. 20, 24, 26

Murphy, K. (2012). *Machine Learning: A Probabilistic Perspective*. Adaptive Computation and Machine Learning series, MIT Press. 29, 30

Nemenyi, P. (1963). *Distribution-free Multiple Comparisons*. Ph.d., Princeton University. 56, 79

Noh, H., You, T., Mun, J. & Han, B. (2017). Regularizing deep neural networks by noise: Its interpretation and optimization. *31st International Conference on Neural Information Processing Systems*. 25

PALMA Neto, L.G. (2004). *Redes Neurais Construtivas para Classificação de Padrões*. Ph.D. thesis, Universidade Federal de São Carlos. 19

Piotrowski, A.P. & Napiorkowski, J.J. (2013). A comparison of methods to avoid overfitting in neural networks training in the case of catchment runoff modelling. *Journal of Hydrology*, **476**, 97–111. 26, 32

Poggio, T. & Girosi, F. (1990). Networks for approximation and learning. *Proceedings of the IEEE*, **78**, 1481 – 1497. 31, 40

Pohlert, T. (2014). *The Pairwise Multiple Comparison of Mean Ranks Package (PMCMR)*. R package. 80

Prechelt, L. (2012). Early stopping - But when? *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **7700 LECTURE NO**, 53–67. 29

R Core Team (2019). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. 79

Reed, R. (1993). Pruning algorithms-a survey. *IEEE Transactions on Neural Networks*, **4**, 740–747. 19

Rifai, S., Glorot, X., Bengio, Y. & Vincent, P. (2011). Adding noise to the input of a model trained with a regularized objective. *arXiv preprint arXiv:1104.3250*. 21

Schlkopf, B., Smola, A.J. & Bach, F. (2018). *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. The MIT Press. 19

Silva, I.B.V.d. & Adeodato, P.J.L. (2011). Pca and gaussian noise in mlp neural network training improve generalization in problems with small and unbalanced data sets. In *The 2011 International Joint Conference on Neural Networks*, 2664–2669. 21

Silvestre, L.J. (2015). Regularização de extreme learning machines: Uma abordagem com matrizes de afinidade. 49

Silvestre, L.J., Lemos, A.P., Braga, J.P. & Braga, A.P. (2015). Dataset structure as prior information for parameter-free regularization of extreme learning machines. *Neurocomputing*, **169**, 288–294. 35, 50, 55

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, **15**, 1929–1958. 25

Stapor, K. (2017). Evaluating and comparing classifiers: Review, some recommendations and limitations. In editor, ed., *CORES 2017: Proceedings of the 10th International Conference on Computer Recognition Systems*, vol. 578, Springer, Cham. 79

Sum, J. & Leung, C.S. (2021). Regularization effect of random node fault/noise on gradient descent learning algorithm. *IEEE Transactions on Neural Networks and Learning Systems*, 1–14. 24

Sum, J., Leung, C.S. & Ho, K. (2012). Convergence analyses on on-line weight noise injection-based training algorithms for MLPs. *IEEE Transactions on Neural Networks and Learning Systems*, **23**, 1827–1840. 27

Tanaka, D., Ikami, D., Yamasaki, T. & Aizawa, K. (2018). Joint optimization framework for learning with noisy labels. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 24

Torres, L.C.B. (2016). Classificador por arestas de suporte (clas): Métodos de aprendizado baseados em grafos de gabriel. 42, 51, 82

Torres, L.C.B., Castro, C.L., Coelho, F., Sill Torres, F. & Braga, A.P. (2015). Distance-based large margin classifier suitable for integrated circuit implementation. *Electronics Letters*, **51**, 1967–1969. 20, 33

Torres, L.C.B., Castro, C.L., Coelho, F. & Braga, A.P. (2020). Large margin gaussian mixture classifier with a gabriel graph geometric representation of data set structure. *IEEE Transactions on Neural Networks and Learning Systems*, 1–7. 41

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, 2nd edn. 36, 37

Wang, C. & Principe, J. (1999). Training neural networks with additive noise in the desired signal. *IEEE Transactions on Neural Networks*, **10**, 1511–1517. 24, 27

Wang, J., Lu, S., Wang, S.H. & Zhang, Y.D. (2021a). A review on extreme learning machine. *Multimedia Tools and Applications*. 35

Wang, Y., Xu, B., Kwak, M. & Zeng, X. (2021b). A noise injection strategy for graph autoencoder training. *Neural Computing and Applications*, **33**, 4807–4814. 26

Yin, S., Liu, C., Zhang, Z., Lin, Y., Wang, D., Tejedor, J., Zheng, T.F. & Li, Y. (2015). Noisy training for deep neural networks in speech recognition. *EURASIP Journal on Audio, Speech, and Music Processing*, **2015**, 2. 25

You, Z., Ye, J., Li, K., Xu, Z. & Wang, P. (2019). Adversarial noise layer: Regularize neural network by adding noise. In *2019 IEEE International Conference on Image Processing (ICIP)*, 909–913. 25

Zhang, M.L. & Zhou, Z.H. (2006). Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering*, **18**, 1338–1351. 40

Zheng, S., Song, Y., Leung, T. & Goodfellow, I.J. (2016). Improving the robustness of deep neural networks via stability training. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4480–4488. 25

Zheng, W., Qian, Y. & Lu, H. (2013). Text categorization based on regularization extreme learning machine. *Neural Computing and Applications*, **22**, 447–456. 36

Zur, R.M., Jiang, Y., Pesce, L.L. & Drukker, K. (2009). Noise injection for training artificial neural networks: A comparison with weight decay and early stopping. *Medical Physics*, **36**, 4810–4818. 20, 24

Apêndice A

Apêndice

A.1 Duas gaussianas

A Figura A.1 sinaliza as referências obtidas com o grafo de Gabriel e apresenta os vetores de borda e os pontos médios na base sintética formada por duas funções gaussianas sem sobreposição.

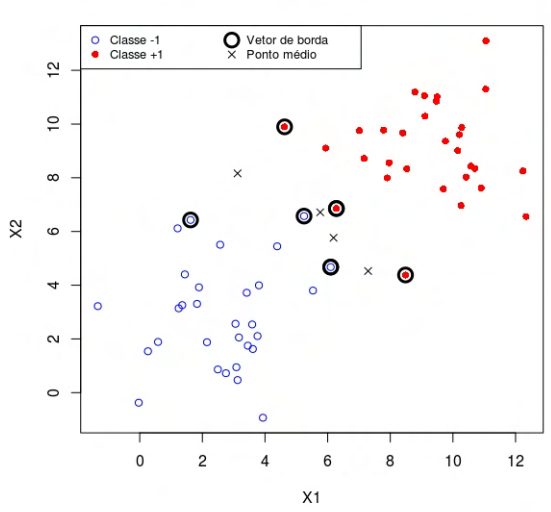


Figura A.1: Indicação dos vetores de borda e os pontos médios.

A.1.1 Ruídos entorno do ponto médio

A superfície de separação, Figura A.2, foi obtida com a adição de cinco amostras sintéticas em cada uma das quatro hiperesferas. O ponto de referência da função geradora são os pontos médios. A norma dos pesos $||\mathbf{w}|| = 1.5578$ e com a média dos erros quadráticos $MSE = 0$.

- **Relação da norma dos pesos com o número de amostras sintéticas**: Ambos os modelos de ajuste dos pesos com regularização e sem regularização mostram a diminuição da norma dos pesos com o aumento das amostras sintéticas no conjunto

de treinamento. O modelo com regularização apresentou-se com menor norma dos pesos em relação ao modelo sem o termo de regularização, ou seja, diminuiu a complexidade do modelo, (ver Figura A.3).

- **Relação do erro quadrático médio pelo número de amostras sintéticas**: Em ambos modelos o acréscimo de amostras sintéticas no conjunto de treinamento fez com que aumentasse o erro médio quadrático. A degradação iniciou após a adição de mais de cinco amostras para o modelo com regularização e após quatorze amostras do modelo sem regularização, (ver Figura A.4).

- **Relação da média da variância nas hiperesferas pelo número de amostras sintéticas**: a intenção é avaliar a dispersão das amostras sintéticas nas hiperesferas para poder determinar a quantidade de amostras sintéticas necessária para atingir a solução ótima do modelo. Vimos que a partir de cinco amostras sintéticas por hiperesfera a oscilação dos valores de variância diminuiu, (ver Figura A.5).

- **Relação da variância por classes pelo número de amostras sintéticas**: A variância em ambas as classes aumentam com o aumento do número de amostras sintéticas adicionadas. Após aproximadamente a adição de cinco amostras a variância começa a diminuir, (ver Figura A.6).

Neste caso o melhor modelo foi selecionado com a adição de poucas amostras sintéticas no conjunto de treinamento, cinco amostras. Em duas situações o modelo sem regularização, com dez e quatorze amostras sintéticas obteve $MSE = 0$ e apresentou $||\mathbf{w}|| = 1.599198$ e $||\mathbf{w}|| = 1.592026$, respectivamente. E o modelo com regularização com cinco amostras sintéticas obteve $||\mathbf{w}|| = 1.5578$ e $MSE = 0$.

A.1.2 Adição de vetores simétricos

A superfície de separação, (Figura A.7), foi obtida com a adição de vinte e nove amostras sintéticas em cada uma das quatro hiperesferas. O ponto de referência da função geradora são os pontos médios e neste caso são geradas amostras de simétricas. A norma dos pesos foi de $||\mathbf{w}|| = 1.4073$ e com a média dos erros quadráticos $MSE = 0$.

- **Relação da norma dos pesos com o número de amostras sintéticas**: Ambos os modelos de ajuste dos pesos com regularização e sem regularização mostram a diminuição da norma dos pesos com o aumento das amostras sintéticas no conjunto de treinamento. O modelo com regularização apresentou-se com menor norma dos pesos em relação ao modelo sem o termo de regularização, ou seja, diminuiu a complexidade do modelo, (ver Figura A.8).

- **Relação do erro quadrático médio pelo número de amostras sintéticas**: Em ambos modelos o acréscimo de amostras sintéticas no conjunto de treinamento fez com que a MSE se mantivesse em zero. Portanto o ajuste do modelo foi realizado dentro da margem de separação, (ver Figura A.9).

- **Relação da média da variância nas hiperesferas pelo número de amostras sintéticas**: com a adição de amostras sintéticas nas hiperesferas houve redução da variância. E a oscilação da variância começa a reduzir após a adição de dez amostras sintéticas nas hiperesferas, (ver Figura A.10).

- **Relação da variância por classes pelo número de amostras sintéticas**: A variância em ambas as classes aumentam no início da adição de amostras sintéticas cerca de cinco por hiperesfera. Após o pico da variância das classes próximo de cinco amostras sintéticas a variância se torna decrescente com o aumento do número de amostras sintéticas, (ver Figura A.11).

Neste caso o melhor modelo foi selecionado com a adição de grande número de amostras sintéticas no conjunto de treinamento, vinte e nove amostras. Como existe simetria em relação as amostras sintéticas ruidosas, o número de amostras em cada hiperesfera dobrou. Por exemplo, o modelo foi selecionado com a adição de vinte e nove amostras, mas o total em cada hiperesfera é de cinquenta e oito amostras.

A.1.3 Ruídos entorno dos vetores de borda

A superfície de separação,(Figura A.12), foi obtida com a adição de uma amostra sintética em cada uma das quatro hiperesferas. O ponto de referência da função geradora são os vetores de borda. A norma dos pesos foi de $||\mathbf{w}|| = 1.6662$ e com a média dos erros quadráticos $MSE = 0$.

- **Relação da norma dos pesos com o número de amostras sintéticas**: Ocorre o aumento da norma dos pesos em ambos os modelos com regularização e sem regularização. Com o aumento do número de amostras sintéticas ruidosas a norma do modelo com regularização foi maior. Neste caso não obtivemos o resultado esperado que é a diminuição da complexidade da rede, ver (Figura A.13).

- **Relação do erro quadrático médio pelo número de amostras sintéticas**: Em ambos modelos o acréscimo de amostras sintéticas no conjunto de treinamento fez com que a MSE ficasse oscilando, apesar da variação ser mínima, $\triangle MSE \approx 0.004$, (ver Figura A.14).

- **Relação da média da variância nas hiperesferas pelo número de amostras sintéticas**: com a adição de amostras sintéticas no entorno dos vetores de borda houve redução da variância, (ver Figura A.15).

- **Relação da variância por classes pelo número de amostras sintéticas**: A variância em ambas as classes aumentam no início da adição de amostras sintéticas. Após o pico da variância das classes próximo de cinco amostras sintéticas a variância se torna decrescente com o aumento do número de amostras, (ver Figura A.16).

A adição de amostras sintéticas entorno dos vetores de borda aumentou a complexidade do modelo, aumento da norma dos pesos. Desta forma o modelo selecionado foi com a adição de uma amostra sintética.

A.1.4 Ruídos entre o ponto médio e o vetor de borda

A superfície de separação, (Figura A.17), foi obtida com a adição de uma amostra sintética em cada uma das quatro hiperesferas. O ponto de referência da função geradora está entre o ponto médio e o vetor de borda. A norma dos pesos foi de $||\mathbf{w}|| = 1.645$ e com a média dos erros quadráticos $MSE = 0$.

- **Relação da norma dos pesos com o número de amostras sintéticas**: Ocorre o aumento da norma dos pesos em ambos os modelos com regularização e sem regularização. Com o aumento do número de amostras sintéticas ruidosas a norma do modelo com regularização foi maior. Neste caso não obtivemos o resultado esperado que é a diminuição da complexidade da rede, (ver Figura A.18).
- **Relação do erro quadrático médio pelo número de amostras sintéticas**: No modelo com regularização houve uma pequena variação da MSE, $\Delta MSE \approx 0.002$, já no modelo sem regularização MSE se manteve em zero, (ver Figura A.19).
- **Relação da média da variância nas hiperesferas pelo número de amostras sintéticas**: com a adição de amostras sintéticas entre o ponto médio e o vetor de borda houve redução da variância, (ver Figura A.20).
- **Relação da variância por classes pelo número de amostras sintéticas**: A variância em ambas as classes aumentam no início da adição de amostras sintéticas. Após o pico da variância das classes próximo de cinco amostras sintéticas a variância se torna decrescente com o aumento do número de amostras. (ver Figura A.21).

A adição de amostras sintéticas entre os pontos médios e os vetores de borda fizeram com que a complexidade do modelo aumentasse, aumentando a norma dos pesos. Desta forma o modelo selecionado foi com a adição de uma amostra sintética.

A.1.5 Ruídos entre os vetores de borda

A superfície de separação, (Figura A.22), foi obtida com a adição de duas amostras sintéticas em cada uma das quatro hiperesferas. O ponto de referência da função geradora está entre os vetores de borda de cada classe. A norma dos pesos foi de $||\mathbf{w}|| = 1.6145$ e com a média dos erros quadráticos $MSE = 0$.

- **Relação da norma dos pesos com o número de amostras sintéticas**: Ocorre o aumento da norma dos pesos em ambos os modelos com regularização e sem regularização. Com o aumento do número de amostras sintéticas ruidosas a norma do modelo com regularização foi maior. Neste caso não obtivemos o resultado esperado que é a diminuição da complexidade da rede, (ver Figura A.23).
- **Relação do erro quadrático médio pelo número de amostras sintéticas**: Para ambos modelos com regularização e sem regularização o $MSE = 0$, (ver Figura A.24).
- **Relação da média da variância nas hiperesferas pelo número de amostras sintéticas**: com a adição de amostras sintéticas houve diminuição da oscilacão da variância, (ver Figura A.25).
- **Relação da variância por classes pelo número de amostras sintéticas**: A variância em ambas as classes aumentaram com a adição de amostras sintéticas. Após o pico da variância das classes próximo de quinze amostras sintéticas a variância decresce lentamente com o aumento do número de amostras, (ver Figura A.26).

A adição de amostras sintéticas entre os vetores de borda fizeram com que a complexidade do modelo aumentasse, aumentando a norma dos pesos. Desta forma o modelo selecionado foi com a adição de duas amostras sintéticas.

A.1.6 Conclusão

O modelo proposto de ajuste dos pesos foi avaliado em várias configurações de regiões para adição de amostras sintéticas ruidosas no conjunto de treinamento. A diminuição da complexidade da rede com a adição de amostras sintéticas ocorreu nos casos onde foram geradas no entorno do ponto médio das arestas de borda. O melhor resultado foi para o caso de adição de amostras sintéticas no entorno do ponto médio com amostras simétricas. Nos demais casos ocorreu aumento da norma dos pesos da rede com adição das amostras sintéticas.

A.2 Ruídos entorno do ponto médio

- Superfície de separação

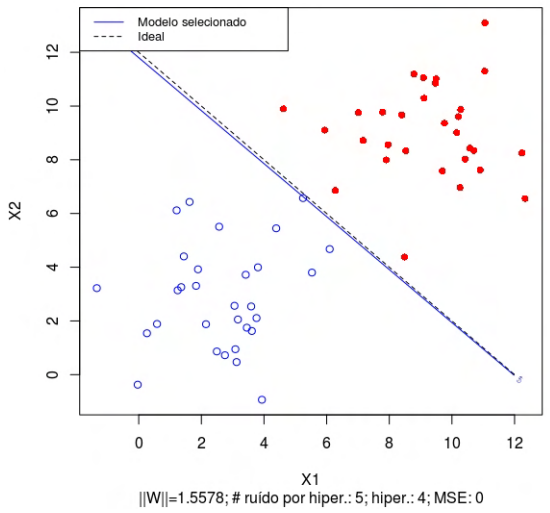


Figura A.2: Superfície de separação.

- Norma dos pesos pela adição de amostras sintéticas ruidosas.

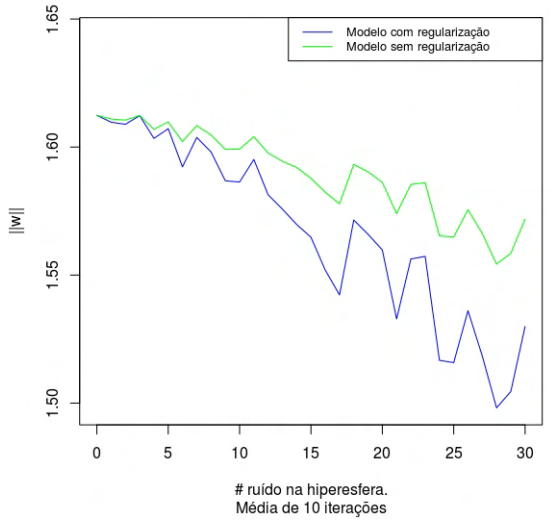


Figura A.3: Relação da norma dos pesos com adição amostras sintéticas

- Relação do erro quadrático médio (MSE) pela adição de amostras sintéticas.

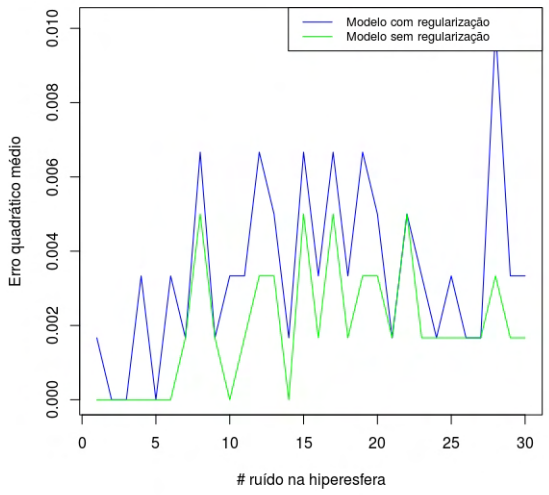


Figura A.4: Relação MSE com adição de amostras sintéticas.

- Relação da média da variância nas hiperesferas com a adição de amostras sintéticas.

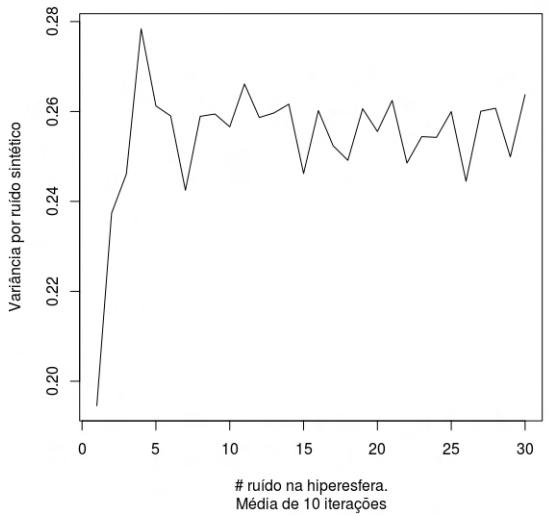


Figura A.5: Relação da média da variância nas hiperesferas com a adição de amostras sintéticas

- Variância por classe em relação a adição de amostras sintéticas nas hiperesferas.

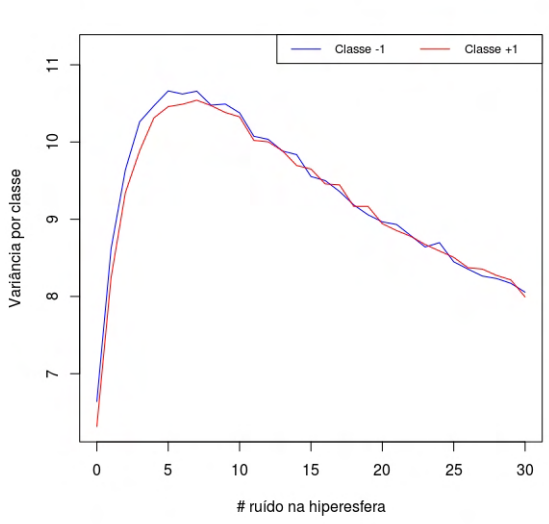


Figura A.6: Relação da média da variância por classe em relação a adição de amostras sintéticas

A.2.1 Adição de vetores simétricos

- Superfície de separação.

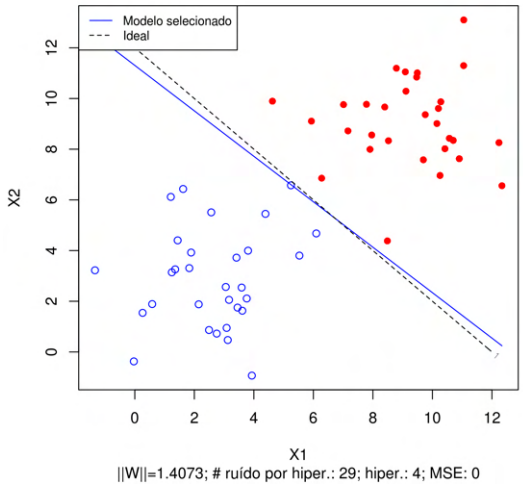


Figura A.7: Superfície de separação

- Norma dos pesos pela adição de amostras sintéticas ruidosas

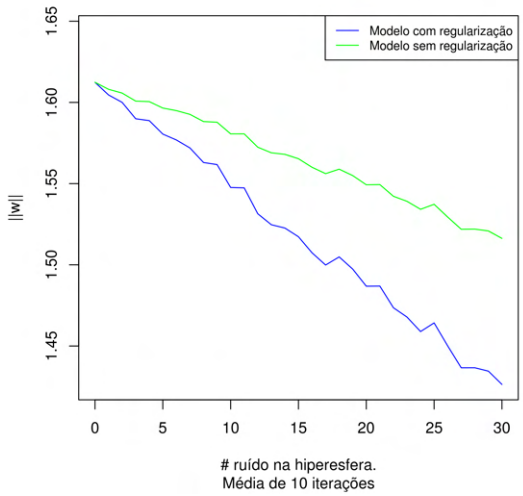


Figura A.8: Relação da norma dos pesos com adição amostras sintéticas

- Relação do erro quadrático médio (MSE) pela adição de amostras sintéticas.

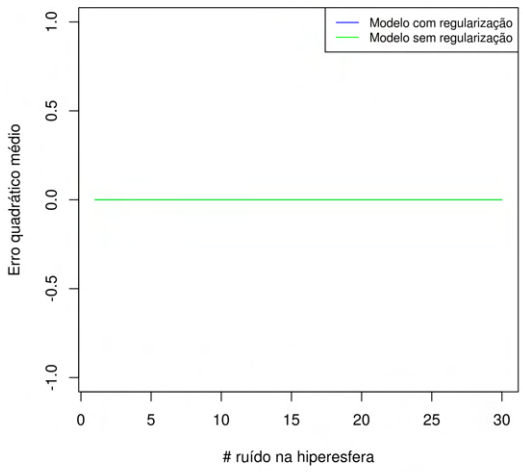


Figura A.9: Relação MSE com adição de amostras sintéticas

- Relação da média da variância nas hiperesferas com a adição de amostras sintéticas.

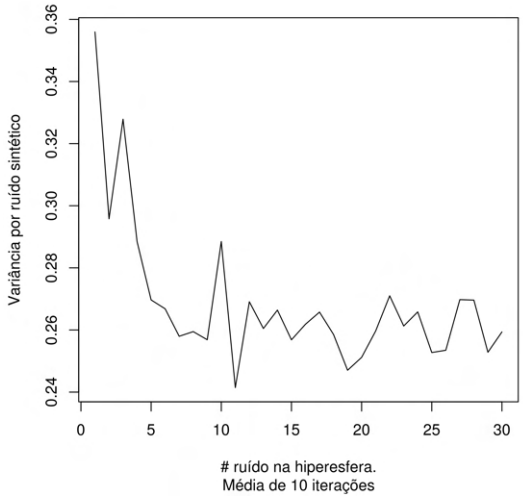


Figura A.10: Relação da média da variância com a adição de amostras sintéticas

- Variância por classe em relação a adição de amostras sintéticas nas hiperesferas.

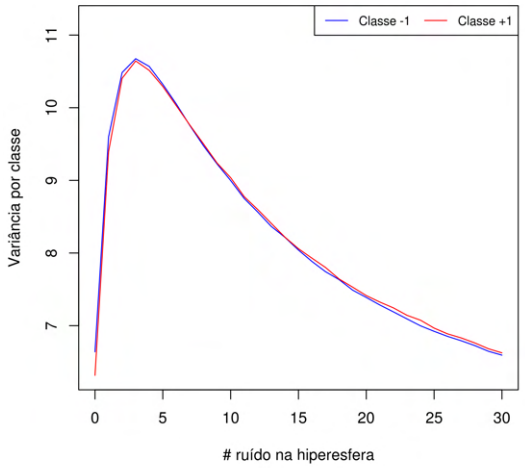


Figura A.11: Relação da média da variância em relação a adição de amostras sintéticas

A.3 Ruídos entorno dos vetores de borda

- Superfície de separação

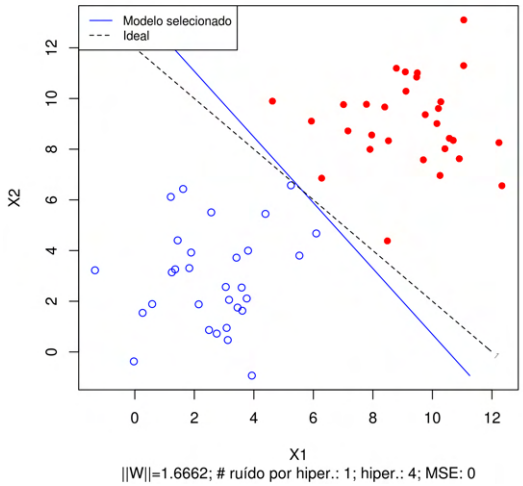


Figura A.12: Superfície de separação

- Norma dos pesos pela adição de amostras sintéticas ruidosas.

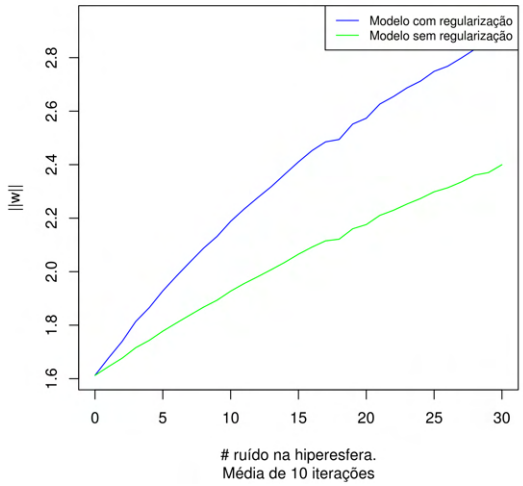


Figura A.13: Relação da norma dos pesos com adição amostras sintéticas

- Relação do erro quadrático médio (MSE) pela adição de amostras sintéticas.

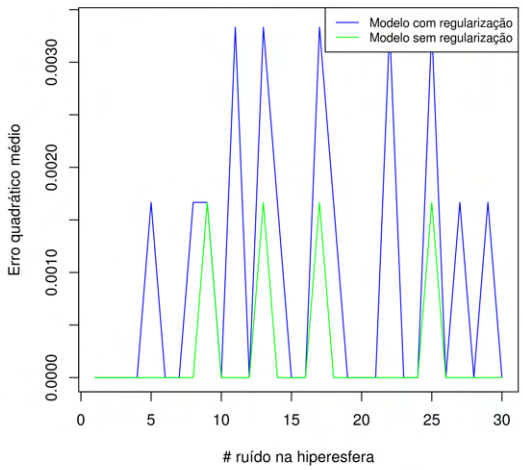


Figura A.14: Relação MSE com adição de amostras sintéticas

- Relação da média da variância nas hiperesferas com a adição de amostras sintéticas.

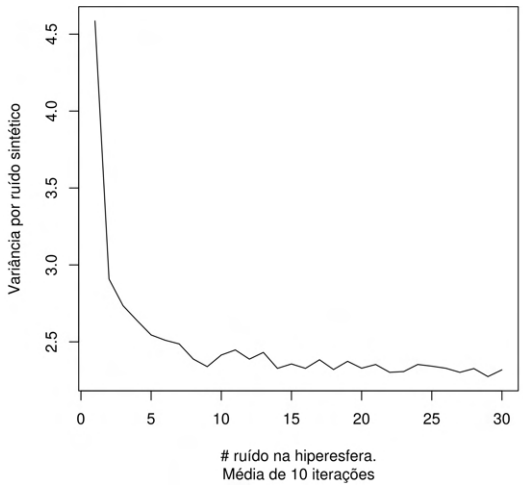


Figura A.15: Relação da média da variância com a adição de amostras sintéticas

- Variância por classe em relação a adição de amostras sintéticas nas hiperesferas.

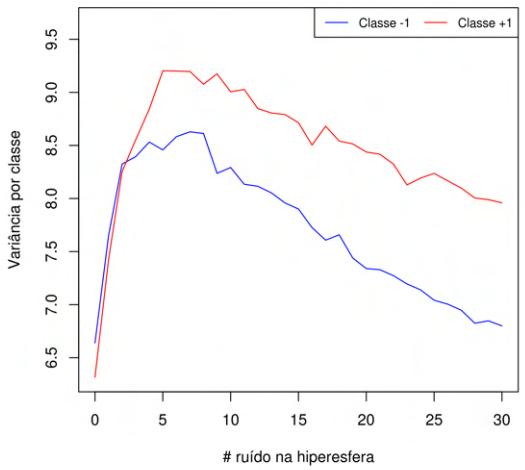


Figura A.16: Relação da média da variância em relação a adição de amostras sintéticas

A.4 Ruídos entre o ponto médio e o vetor de borda

- Superfície de separação.

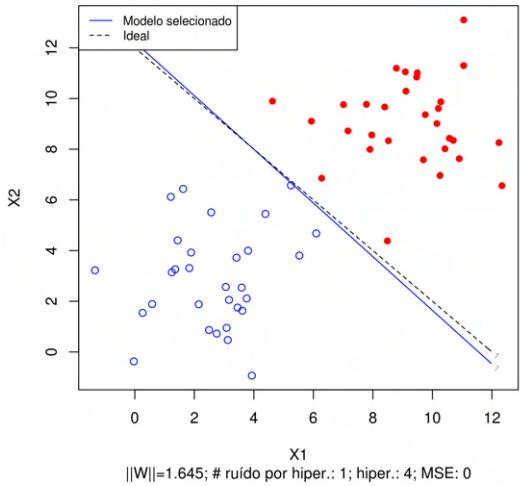


Figura A.17: Superfície de separação

- Norma dos pesos pela adição de amostras sintéticas ruidosas.

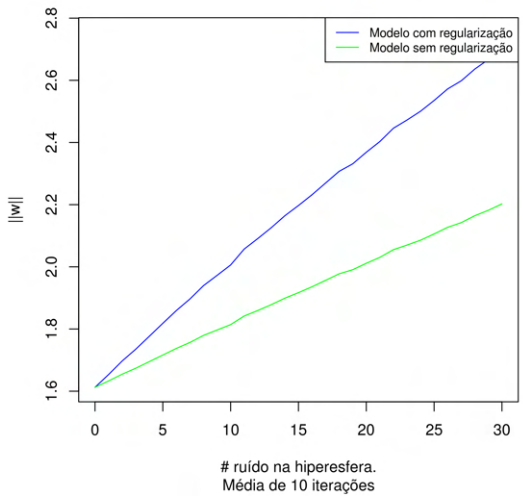


Figura A.18: Relação da norma dos pesos com adição amostras sintéticas

- Relação do erro quadrático médio (MSE) pela adição de amostras sintéticas.

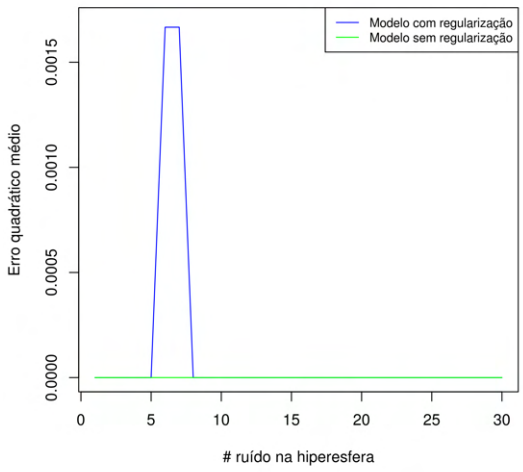


Figura A.19: Relação MSE com adição de amostras sintéticas

- Relação da média da variância nas hiperesferas com a adição de amostras sintéticas.

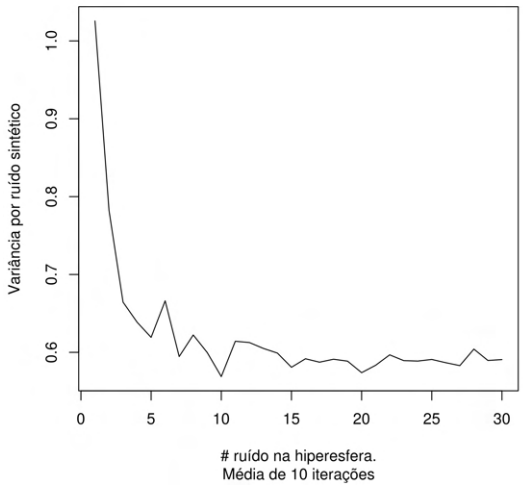


Figura A.20: Relação da média da variância com a adição de amostras sintéticas

- Variância por classe em relação a adição de amostras sintéticas nas hiperesferas.

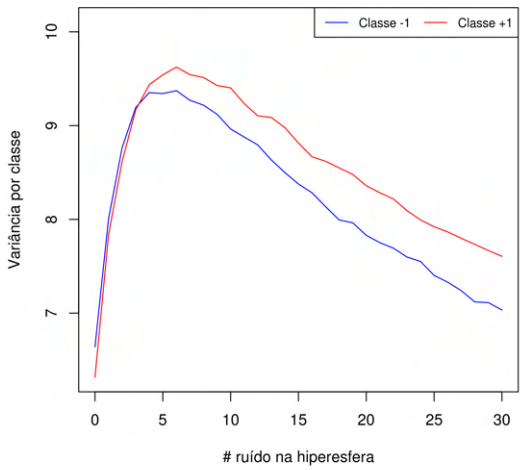


Figura A.21: Relação da média da variância em relação a adição de amostras sintéticas

A.5 Ruídos entre os vetores de borda

- Superfície de separação.

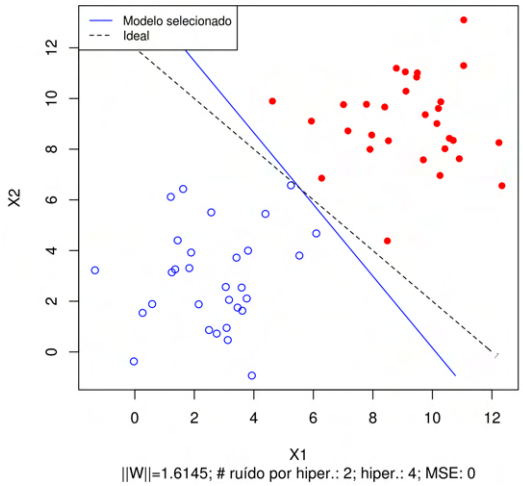


Figura A.22: Superfície de separação

- Norma dos pesos pela adição de amostras sintéticas ruidosas.

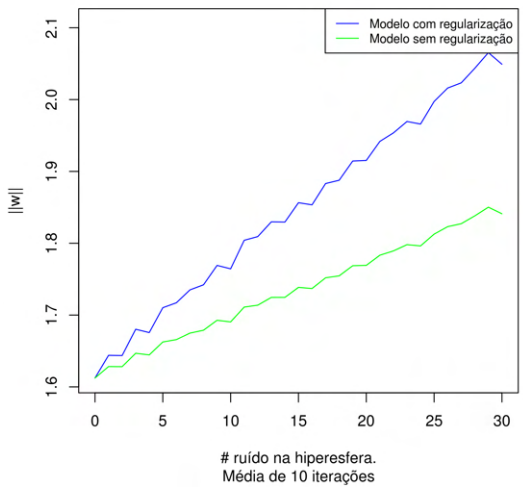


Figura A.23: Relação da norma dos pesos com adição amostras sintéticas

- Relação do erro quadrático médio (MSE) pela adição de amostras sintéticas

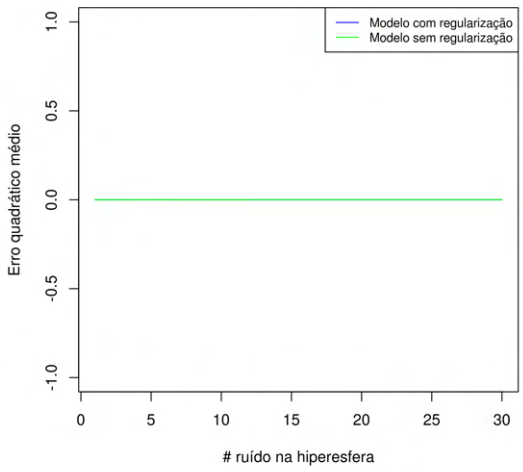


Figura A.24: Relação MSE com adição de amostras sintéticas

- Relação da média da variância nas hiperesferas com a adição de amostras sintéticas.

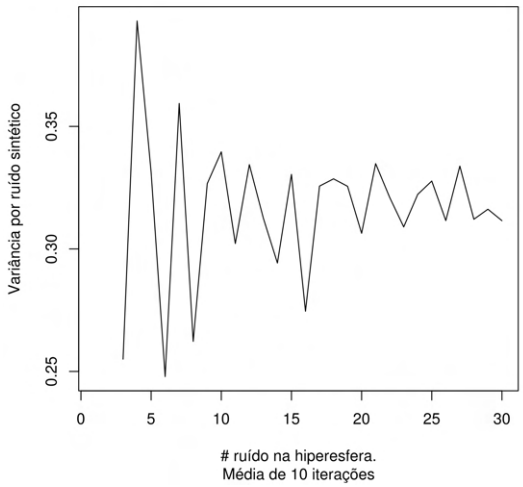


Figura A.25: Relação da média da variância com a adição de amostras sintéticas

- Variância por classe em relação a adição de amostras sintéticas nas hiperesferas.

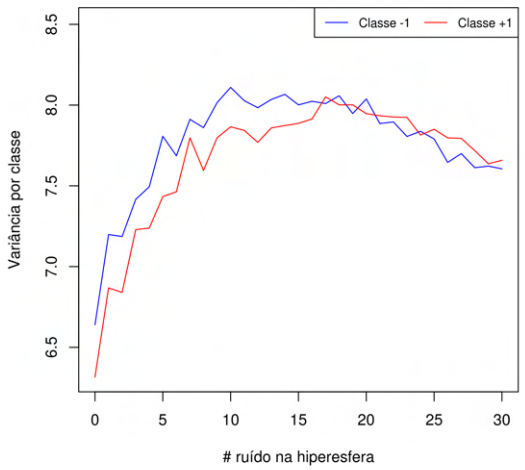


Figura A.26: Relação da média da variância em relação a adição de amostras sintéticas