

UNIVERSIDADE FEDERAL DE MINAS GERAIS
ESCOLA DE ENGENHARIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

PAULO VITOR DE CAMPOS SOUZA

**AVALIAÇÃO DE TÉCNICAS DE REGULARIZAÇÃO APLICADAS AS REDES
NEURAIS NEBULOSAS.**

Belo Horizonte

Ano 2015

PAULO VITOR DE CAMPOS SOUZA

**AVALIAÇÃO DE TÉCNICAS DE REGULARIZAÇÃO APLICADAS AS REDES
NEURAIS NEBULOSAS.**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais para obtenção do grau de Mestre em Engenharia Elétrica.

Linha de Pesquisa: Inteligência Computacional

Orientador: André Paim Lemos

BELO HORIZONTE

ANO 2015

Ficha catalográfica: elaborada pela biblioteca da

Será impressa no verso da folha de rosto e não deverá ser contada.



Universidade Federal de Minas Gerais
Escola de Engenharia
Programa de Pós-Graduação em Engenharia Elétrica

Dissertação intitulada Avaliação de Técnicas de Regularização Aplicadas as Redes Neurais Nebulosas, de autoria de Paulo Vitor de Campos Souza, aprovada pela banca examinadora constituída pelos seguintes professores:

Prof. Dr. André Paim Lemos
Universidade Federal de Minas Gerais

Prof. Dr. Walmir Matos Caminhas
Universidade Federal de Minas Gerais

Prof. Dr. Cristiano leite de Castro
Universidade Federal de Minas Gerais

Prof. Rodney Rezende Saldanha
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica - PPGEE/UFMG.

Data de aprovação: Belo Horizonte, 09 de outubro de 2015

DEDICATÓRIA

Dedico esse trabalho a todos os meus familiares que amo muito e que sempre me deram suporte em todas as dificuldades que passei. Dedico essa dissertação a minha doce mãe Rose, ao meu determinado pai Marcos e a minha linda irmã Michelly. Sem vocês eu tenho certeza que esse trabalho jamais seria concluído.

Dedico também essa dissertação aos meus amigos presentes, que me apoiam e me fazem rir em momentos que passei por dificuldades diversas e também aos meus amigos que não estão mais nesse plano, pois tenho certeza que do lugar que estão, torcem e comemoram comigo todas as vitórias e avanços de minha vida.

AGRADECIMENTOS

Agradeço em primeiro lugar a Deus por todas as glórias e vitórias que tem me propiciado nesses últimos anos.

Agradeço ao meu orientador André Paim pelos excelentes conselhos e pela sua paciência na orientação desse trabalho acadêmico. Sua experiência e competência foram fundamentais para a conclusão de um trabalho com qualidade e de relevância a comunidade acadêmica.

Agradeço aos colegas de mestrado que tanto contribuíram para me acalmar em momentos que tinha dificuldades durante as disciplinas, em especial o colega David por me aconselhar em momentos tão turvos em minha trajetória acadêmica.

Agradeço ao professor Antônio Braga e a sua equipe do LITC por sempre acreditar no meu trabalho e contribuir efetivamente para a minha produção científica.

Agradeço a Companhia de Tecnologia da Informação do Estado de Minas Gerais (PRODEMGE) e aos meus ex- companheiros de empresa e amigos pessoais por estarem sempre ao meu lado participando desse momento de grande realização pessoal.

Agradeço as minhas amigas Mariana e Regiane, ao meu amigo Breno, aos amigos distantes, mas não menos especiais, Anderson e Wagner que sempre acreditaram em meu potencial. Aos amigos que não estão mais entre a gente: Leandro e Adriano que sempre me protegeram e guiaram durante toda a minha vida. Em geral todos os meus amigos que de alguma forma se fizeram presentes.

E por fim um agradecimento mais do que especial ao meu pai, mãe e irmã que comemoram comigo as minhas vitórias, não me deixaram esmorecer ao enfrentar os desafios e estiveram comigo todo esse período de extrema dedicação e luta que foi o meu mestrado. A vocês meu mais sincero e caloroso muito obrigado. Sem minha família eu não sou nada. Nenhum gesto será suficiente para recompensar todo o bem que vocês têm feito por mim.

“No meio da dificuldade encontra-se a
oportunidade.”

Albert Einstein

RESUMO

Este trabalho propõe um novo algoritmo de treinamento para redes neurais nebulosas que é capaz de gerar modelos parcimoniosos e com algum grau de interpretabilidade. Em alguns casos, como em redes neurais nebulosas, o aprendizado pode-se tornar uma tarefa muito complexa e lenta. Nesse trabalho o aprendizado é realizado baseando-se em conceitos de máquinas de aprendizado extremo para estimativa dos parâmetros e em uma técnica de seleção de características baseada em regularização e reamostragem, denominada *bootstrap lasso*, para definição da topologia da rede. A utilização da regularização nas camadas internas do modelo permite que este seja mais preciso em suas respostas e que um conjunto de regras nebulosas seja extraído de sua topologia possibilitando a interpretabilidade dos resultados obtidos. Experimentos numéricos são apresentados para problemas de classificação de padrões utilizando bases reais de pequenas e grandes dimensões. Os resultados obtidos são comparados a outros classificadores de referência na literatura. A análise estatística dos resultados sugere que o algoritmo proposto possui uma acurácia similar a modelos de máquina de aprendizado extremo regularizados, porém com uma topologia interpretável.

Palavras-chave: Redes neurais nebulosas, regularização, *bootstrap lasso*, neurônios lógicos nebulosos.

ABSTRACT

This paper proposes a new training algorithm for fuzzy neural networks that is able to generate parsimonious models with some degree of interpretability. In some cases, as in fuzzy neural networks learning can become a very slow and complex task. In this work learning is performed based on concepts of extreme learning machines to estimate parameters and a feature selection technique based on regularization and resampling called bootstrap lasso, to define the network topology. The use of regularization in the inner layers of the model enables it to be more precise in its answers, and concise set of fuzzy rules can be extracted from the resulting topology allowing the interpretability of the results. Numerical results are presented for pattern classification problems using real bases of large and small dimensions. The results are compared to other classifiers reference in the literature. Statistical analysis of the results suggests that the proposed algorithm has a similar accuracy to regularized extreme machine learning models, but with an interpretable topology.

Keywords: Fuzzy neural networks, regularization, bootstrap Lasso, fuzzy logic neurons

LISTA DE FIGURAS

Figura 1- Neurônio biológico.....	20
Figura 2- Neurônio artificial.....	21
Figura 3- Neurônio nebuloso.....	22
Figura 4- Neurônio nebuloso tipo I.....	23
Figura 5- Neurônio nebuloso tipo II.....	23
Figura 6- Neurônio nebuloso tipo III.....	24
Figura 7- Diagramas de neurônios e suas especializações.....	26
Figura 8- Exemplo de redes <i>feedforward</i> e redes <i>reccurent</i>	28
Figura 9- Topologias de redes neurais nebulosas com três camadas.....	30
Figura 10- Arquitetura simplificada do modelo ANFIS com quatro camadas.....	30
Figura 11- Estrutura da rede neural HyFIS com cinco camadas.....	31
Figura 12- Diagramas de modelos de redes neurais nebulosas diversas e suas especificidades.....	35
Figura 13- Regressão <i>Ridge</i>	37
Figura 14- Coeficientes <i>Ridge</i> como uma função de diminuição do parâmetro λ	38
Figura 15- Regressão lasso.....	39
Figura 16- Imagem Estimativa para o <i>lasso</i> (esquerda) e regressão de <i>ridge</i> (direita). São mostrados os contornos das funções de erro e de restrição.....	39
Figura 17- Contornos dos valores da constante $j \beta_j _q$ conforme o valor de q se altera.....	40
Figura 18- Contornos dos valores da constante $j \beta_j _q$ para $q=1.2$ e $\alpha=0.2$ para valores em $\lambda_j = 1p(\alpha\beta_j^2 + 1-\alpha\beta_j)$	43
Figura 19- Rede Neural Nebulosa Proposta.....	50
Figura 20- Exemplo Rede Neural Nebulosa com $N=2$, $M=3$ e $L=3$	51
Figura 21- Funções de pertinência gaussianas igualmente espaçadas.....	54
Figura 22- Base de dados sintética para problema de classificação.....	59
Figura 23- Funções de Pertinência geradas para a base sintética.....	60
Figura 24- Plot do Teste 2 - Acurácia dos algoritmos analisados.....	63
Figura 25- Validações do Teste ANOVA- Teste 2.....	64
Figura 26- Plot do Teste 3 - Acurácia dos algoritmos propostos.....	66
Figura 27- Validações do Teste ANOVA- Teste 3.....	67

LISTA DE TABELAS

Tabela 1- Informações das Bases de Teste.	61
Tabela 2- Resultados de acurácia nos testes efetuados.....	62
Tabela 3- Neurônios selecionados no teste.....	62
Tabela 4- Resultados das múltiplas comparações de Tukey.....	64
Tabela 5- Resultados das acurácias dos testes executados na comparação entre o modelo regularizado x não regularizado.	65
Tabela 6- Resultado do Teste Post –Hoc de Dunnett.....	68

LISTA DE QUADROS

QUADRO 1 – Mapeamento e características das t-normas e s-normas.....	19
QUADRO 2 – Mapeamento e características das uninormas e nullnormas.	20
QUADRO 3 – Algoritmo Bolasso.....	42
QUADRO 4 – Algoritmo de Aprendizado Proposto para Redes Neurais Nebulosas	57

LISTA DE ABREVIATURAS

ANFIS	–	ADAPTIVE NEURO-FUZZY INFERENCE SYSTEM
ANOVA	–	ANALYSIS OF VARIANCE
BOLASSO OPERATOR	–	BOOTSTRAP-ENHANCED LEAST ABSOLUTE SHRINKAGE
eGNN	–	ENVOLVING GRANULAR NEURAL NETWORK
ePL	–	ENVOLVING PARTICIPATORY LEARNING
ELM	–	EXTREME LEARNING MACHINE
FORELM	–	ONLINE REGULARIZED EXTREME LEARNING MACHINE WITH FORGETTING MECHANISM
FOS-ELM	–	ONLINE SEQUENTIAL ELM WITH FORGETTING MECHANISM
HyFIS	–	HYBRID NEURAL FUZZY INFERENCE SYSTEM
KB-IELM	–	KERNEL BASED INCREMENTAL ELM
LARS	–	LEAST ANGLE REGRESSION
MAX	–	MÁXIMO
MFS	–	MEMBERSHIP FUNCTION
MIN	–	MÍNIMO
NFN	–	NEO FUZZY NEURONS
OP-ELM	–	OPTMALLY PRUNED EXTREME LEARNING MACHINE
Re-OS-ELM	–	REGULARIZED ONLINE SEQUENTIAL ELM
RNA	–	REDE NEURAL ARTIFICIAL
SLFN	–	SINGLE HIDDEN-LAYER FEEDFORWARD NEURAL NETWORKS

SUMÁRIO

1 INTRODUÇÃO	14
1.1 PROBLEMA	15
1.2 OBJETIVOS	15
1.2.1 Objetivos específicos.....	15
1.3 JUSTIFICATIVA	16
1.4 CONTRIBUIÇÕES DO TRABALHO	16
1.5 ESTRUTURA DA DISSERTAÇÃO	16
2 CONCEITOS GERAIS E REVISÃO DA LITERATURA	17
2.1 REDES NEURAIS ARTIFICIAIS	17
2.2 SISTEMAS NEBULOSOS	17
2.2.1 Conceitos referentes a conjuntos nebulosos.....	18
2.3 NEURÔNIO NEBULOSO GENÉRICO	20
2.4 CLASSIFICAÇÃO DOS NEURÔNIOS NEBULOSOS	22
2.5 REDES NEURAIS NEBULOSAS	26
2.6 CARACTERÍSTICAS DE REDES NEURAIS NEBULOSAS.....	27
2.7 MÉTODOS DE REGULARIZAÇÃO.....	35
2.8 REGRESSÃO <i>RIDGE</i>	36
2.9 MÉTODO <i>LASSO</i>	38
2.9.1 Bolasso.....	41
2.10 <i>ELASTIC NET</i>	42
2.11 REGULARIZAÇÃO MULTIOBJETIVO	43
2.12 MODELOS REGULARIZADOS	45
3 REDE NEURAL NEBULOSA PARA CLASSIFICAÇÃO BINÁRIA	49
3.1 NEURÔNIOS LÓGICOS NEBULOSOS UTILIZADOS	49
3.2 REDE NEURAL NEBULOSA.	50

3.3 TREINAMENTO DA REDE NEURAL NEBULOSA UTILIZANDO MÁQUINA DE APRENDIZADO EXTREMO.....	52
3.3.1 Geração de funções de pertinência.....	54
3.3.2 Procedimento de regularização da rede.....	55
3.3.3 Atualização de pesos da rede neural artificial de agregação e utilização da Máquina de Aprendizado Extremo	56
4 EXPERIMENTOS NUMÉRICOS	58
4.1 CARACTERÍSTICAS DO PROBLEMA E FORMA DE AVALIAÇÃO	58
4.2 CARACTERÍSTICAS GERAIS DOS ALGORITMOS E TESTES EXECUTADOS.	58
4.3 TESTES DE INTERPRETABILIDADE DA REDE RESULTANTE	59
4.3.1 Base de dados sintética	59
4.3.2 Resultados experimentais	60
4.4 TESTES DE APERFEIÇOAMENTO DA ESTRUTURA DA REDE.....	60
4.4.1 Base de dados reais.....	60
4.4.2 Aperfeiçoamento da rede resultante, resultados experimentais e análises estatísticas.	61
4.5 TESTES DE EFICIÊNCIA DO MODELO DE TREINAMENTO REGULARIZADO.....	65
4.5.1 Resultados experimentais e análises estatísticas.	65
5 CONCLUSÕES	69
REFERÊNCIAS.....	71

1 Introdução

A área inteligência computacional apresenta grandes avanços no desenvolvimento de técnicas e modelos que simulam processos e sistemas, com destaque para redes neurais artificiais, sistemas nebulosos e seus modelos híbridos, com grande quantidade de novas aplicações sendo propostas na literatura. Um dos principais objetivos das pesquisas em inteligência computacional é criar e modelar sistemas computacionais que emulem características específicas dos seres humanos, como aprendizado, intuição, raciocínio lógico, classificação e regressão. (Haykin, 1994).

A partir da experiência vivida pelo indivíduo, onde o mesmo é capaz de visualizar um objeto e dizer do que se trata, surgiram estudos para realizar a simulação dessa atitude computacionalmente. Essa característica de reconhecimento humano é simulada em modelos computacionais através de manipulações matemáticas. Para realizar suas atividades os modelos necessitam de informações de entrada para que realizem suas tarefas de classificar, atribuindo a eles um rótulo, nome ou condição (Tou e Gonzalez, 1981).

Um dos modelos matemáticos utilizados para realizar algumas simulações do comportamento humano (como classificar padrões, prever, etc.) são as redes neurais nebulosas, que são redes neurais compostas por neurônios nebulosos. Esses modelos diferem dos modelos tradicionais de redes neurais e de sistemas nebulosos devido à cooperação entre os conceitos de redes neurais com as ideias oriundas da teoria dos conjuntos nebulosos, gerando um modelo capaz de utilizar a capacidade de aprendizado das redes neurais em conjunto a eficiência do trato a imprecisão e a interpretação que possuem os sistemas nebulosos (Pedrycz, 1991).

As redes neurais nebulosas utilizam-se, em grande parte dos modelos, da estrutura de uma rede neural artificial, onde os neurônios artificiais clássicos são substituídos por neurônios nebulosos (Pedrycz, 1993). Sua principal característica é a transparência, permitindo a utilização de informação a priori para definição da arquitetura inicial da rede e a extração de informações valiosas a partir da topologia resultante. Assim, a rede neural passa a ser vista como um sistema linguístico, preservando a capacidade de aprendizado da RNA. Seus neurônios nebulosos são compostos de normas triangulares, que generalizam as operações de união e interseção de conjuntos clássicos para a teoria de conjuntos nebulosos. Como exemplos de neurônios nebulosos destacam-se os neurônios

and e *or*, onde o seu tipo baseia-se na maneira com que as entradas e os pesos são agregados. (Hell, 2008).

Recentemente, ideias provenientes da teoria de máquinas de aprendizado extremo (ELM) (Huang et al, 2006) vem sendo utilizadas para o treinamento de redes neurais nebulosas (Lemos et al, 2012). Essa motivação se deve principalmente ao desempenho e baixo custo computacional apresentado por treinamentos baseados em ELMs

1.1 Problema

Problemas para selecionar a quantidade de neurônios ideais em redes neurais artificiais e em redes neurais nebulosas são comuns em modelos de aprendizagem de máquina. Quando um modelo possui quantidade excessiva de neurônios em sua estrutura a resposta final da rede pode ser afetada devido à especialização demasiada em relação ao conjunto de treinamento. Esse comportamento pode gerar um problema de aprendizado como o super aprendizado (*over-training/over-fitting*). Selecionar a quantidade ideal de neurônios que compõem uma rede neural nebulosa pode ser uma tarefa complexa, devido às várias características do modelo. Além disso, não é possível realizar a extração de informações de contexto do problema pelas ações realizadas por modelos inteligentes. A falta de interpretabilidade de um resultado baseado na metodologia utilizada é um fator que dificulta a utilização de redes neurais nebulosas.

1.2 Objetivos

O objetivo desse trabalho é desenvolver um modelo matemático computacional baseado nos conceitos de redes neurais artificiais e sistemas nebulosos para classificação binária, propondo métodos de poda na arquitetura da camada interna da rede neural nebulosa utilizando uma metodologia de seleção de regras mais relevantes através de técnicas de regularização, possibilitando a interpretabilidade do modelo.

1.2.1 Objetivos específicos

Como objetivos específicos, é proposto estudar os métodos de granularização do espaço de entrada, metodologias de regularização e poda executadas na camada intermediária de modelos, métodos de seleção de regras através de sua relevância ao modelo e buscar a melhor alternativa para melhorar a acurácia de modelos de redes neurais nebulosas ao realizar a classificação de padrões.

1.3 Justificativa

Como modelos inteligentes estão cada vez mais presentes no cotidiano de empresas e pessoas, a necessidade que esses modelos tenham a resposta mais próxima ao real é necessária. A relevância desse estudo busca trazer a melhoria da resposta de um modelo de redes neurais nebulosas a partir da seleção de regras com maior relevância, objetivando um modelo mais preciso, interpretável e que possa ser usufruído pela sociedade dentro de vários contextos em seus problemas cotidianos.

1.4 Contribuições do Trabalho

Um trabalho referente a esse texto foi submetido e aprovado em um congresso. Ele é apresentado a seguir:

SOUZA, P.V.C, Lemos, A.P. Redes Neurais Nebulosas para problemas de Classificação. In: Simpósio Brasileiro de Automação Inteligente, 2015, Natal-RN. XII SBAI-Simpósio Brasileiro de Automação Inteligente.

1.5 Estrutura da dissertação

O restante da dissertação está organizado da seguinte forma. O Capítulo 2 apresenta os conceitos teóricos relativos a redes neurais, sistemas nebulosos, redes neurais nebulosas, os neurônios lógicos nebulosos em suas várias versões, exemplos de modelos de redes neurais nebulosas e conceitos sobre regularização. No Capítulo 3 é descrita a metodologia proposta de treinamento das redes neurais nebulosas. No Capítulo 4 resultados de experimentos numéricos são apresentados. Finalmente, no Capítulo 5 são apresentadas as conclusões.

2 Conceitos gerais e revisão da literatura

Neste capítulo serão tratados conceitos do “estado da arte” sobre o tema a que se refere o estudo dessa dissertação. São apresentados conceitos referentes às redes neurais artificiais e todos os elementos que compõem seu funcionamento. Os sistemas nebulosos e os modelos híbridos formados a partir desses dois conceitos têm fundamental importância nesse capítulo. Ainda são apresentados conceitos sobre regularização e várias formas de realizá-la.

2.1 Redes Neurais Artificiais

Uma rede neural de multicamadas composta por uma camada de entrada, uma ou mais camadas escondidas e uma camada de saída. A rede pode ser completamente conectada onde cada neurônio está ligado a todos os neurônios da camada seguinte, parcialmente conectada onde cada neurônio se encontra ou então localmente conectada onde há uma conexão parcial orientada para cada tipo de funcionalidade. Para realizar o treinamento de uma rede neural é necessário um conjunto de dados que contenha padrões para o treinamento e saídas desejadas. Dessa forma o problema do treinamento das redes neurais se resume em um problema de otimização em que se deseja encontrar o melhor conjunto de pesos que minimize o erro médio quadrático calculado entre as saídas da rede e as saídas desejadas (Braga et al, 2007).

Para Haykin (2001) uma rede neural é um processador distribuído e em paralelo constituído de unidades de processamento menos complexas, com finalidade de armazenar o conhecimento acerca de um tema ou sobre um conjunto de características, tornando-o disponível para uso em tarefas similares ao dos seres humanos. Como as redes neurais artificiais buscam simular o comportamento do cérebro humano, o conhecimento é adquirido pela rede através do ambiente que o cerca, por meio de um processo de aprendizado. Já no armazenamento do conhecimento, a analogia é feita com as forças de conexões entre os neurônios, conhecido como pesos sinápticos.

2.2 Sistemas Nebulosos

Os sistemas nebulosos são baseados na lógica nebulosa, desenvolvida por Zadeh em 1965. Seu trabalho foi motivado devido a grande variedade de informações vagas e incertas na tomada de decisões humanas. Alguns problemas não podem ser resolvidos

com a lógica padrão booleana, dado que, em algumas situações, somente dois valores são insuficientes para resolver um problema (Hell, 2008).

Para Calvo (2007) o uso dos sistemas nebulosos faz-se necessários em casos onde a abordagem clássica torna-se inviável para a resolução de um problema devido à sua complexidade. Os métodos mais conhecidos são passíveis de alterações bruscas para a resolução de problemas devido à simplificação do modelo real, porém os sistemas nebulosos possuem recursos (funções de pertinência, regras e operadores de agregação) que possibilitam a aproximação mais fiel aos dados, evitando que a solução gerada pelo sistema nebuloso destoe consideravelmente do problema real.

2.2.1 Conceitos referentes a conjuntos nebulosos

Diferentemente dos conjuntos clássicos, o conjunto nebuloso introduzido por Zadeh (1965) pode ser definido como um conjunto impreciso, onde a transição entre pertencer ou não a um conjunto avaliado ocorre de forma gradual ao invés de uma forma brusca. Ao definir um conjunto nebuloso F em um universo de discurso X estamos caracterizando uma função de pertinência F que realiza a associação de cada elemento x qualquer, onde $x \in X$, com um grau de pertinência $F(x)$ que pertence ao intervalo entre zero e um, podendo ser representado por $F(x) \in [0,1]$. Esse conjunto nebuloso F em X também pode ser representado por um conjunto de pares ordenados na forma de $\{(x, F(x)) \mid x \in X\}$ (Hell, 2008).

Assim como são realizadas com conjuntos clássicos, os conjuntos nebulosos também podem realizar operações de união e interseção. Porém, diferentemente da forma que são realizadas nos conjuntos clássicos, usando operadores *min* e *max*, os conjuntos nebulosos utilizam normas triangulares (t-norma) e co-normas triangulares (t-conorma ou s-norma) respectivamente. (Hell, 2008).

Para Yager & Ribalov (1996) os conceitos de t-norma e s-norma desempenham um papel importante na generalização dos operadores de agregação AND e OR. Esses operadores, definidos no intervalo unitário, distinguem-se uns dos outros pelos seus elementos de identidade. Estes elementos, um para a t-norma e zero para o s-norma, não afetam a agregação de valor.

O Quadro 1 a seguir expõe e compara as características de uma t-norma e uma s-norma, evidenciando o mapeamento e as propriedades de ambas.

QUADRO 1 – Mapeamento e características das t-normas e s-normas

Normas Triangulares	
t-norma (S)	s-norma (S)
Mapeamento: $T: [0, 1] \times [0, 1] \rightarrow [0, 1]$	Mapeamento: $S: [0, 1] \times [0, 1] \rightarrow [0, 1]$
Características	Características
(1) $T(a, b) = T(b, a)$ (Comutatividade)	(1) $S(a, b) = S(b, a)$ (Comutatividade)
(2) $T(a, b) \geq T(c, d)$ se $a \geq c$ e $b \geq d$ (Monotonicidade)	(2) $S(a, b) \geq S(c, d)$ se $a \geq c$ e $b \geq d$ (Monotonicidade)
(3) $T(a, T(b, c)) = T(T(a, b), c)$ (Associatividade)	(3) $(S, S(b, c) = S(S(a, b), c)$, (Associatividade)
(4) $T(a, 1) = a$ (Limite)	(4) $S(a, 0) = a$ (Limite)
(5) Elemento Neutro=1	(5) 1= Elemento Neutro=0

Fonte: Adaptado de (Yager & Ribalov, 1996).

Podemos verificar no Quadro 1 que os elementos neutros são restritos (1 para t-norma e 0 para s-norma). Buscando resolver esse problema a fim de generalizar as t-normas e s-normas foram criados operadores que relaxam as restrições correspondentes aos elementos neutros, permitindo que esse elemento possua valores no intervalo unitário (Hell, 2008). Yager e Ribalov provaram que é possível realizar o processo de generalização das normas triangulares, unindo as t-normas e s-normas, chamando essa unificação de operadores como uninorma (Yager & Ribalov, 1996).

A uninorma tem as mesmas três primeiras propriedades como a t-norma e s-norma, mas permite maiores opções em relação à propriedade 4 listada no Quadro 1. Indicamos que o operador satisfaz a quarta propriedade atribuindo-lhe um elemento identidade $g \in [0, 1]$. Vemos, então, que a t-norma é um caso especial de uninorma com $g = 1$, enquanto o s-norma é um caso especial com $g = 0$ (Yager & Ribalov, 1996). Hell (2008) ainda complementa dizendo que com a opção do elemento identidade selecionado entre o intervalo $[0, 1]$, uma uninorma pode variar suavemente entre um *and* puro para um *or* puro quando esse elemento identidade varia de 1 para 0.

Já a nullnorma pode ser definida como um operador binário $NU: [0, 1]^2 \rightarrow [0, 1]$ que é cumulativo, monotônico, associativo e possui um elemento $e \in [0, 1]$ chamado de elemento de absorção. Fazendo a analogia com a uninorma, quando o elemento de absorção é zero, NU é utilizado como uma t-norma, ao mesmo tempo em que quando o

valor de $e = 1$ utiliza-se NU como s-norma (Hell, 2008). O Quadro 2 apresenta as principais características das uninormas e nullnormas.

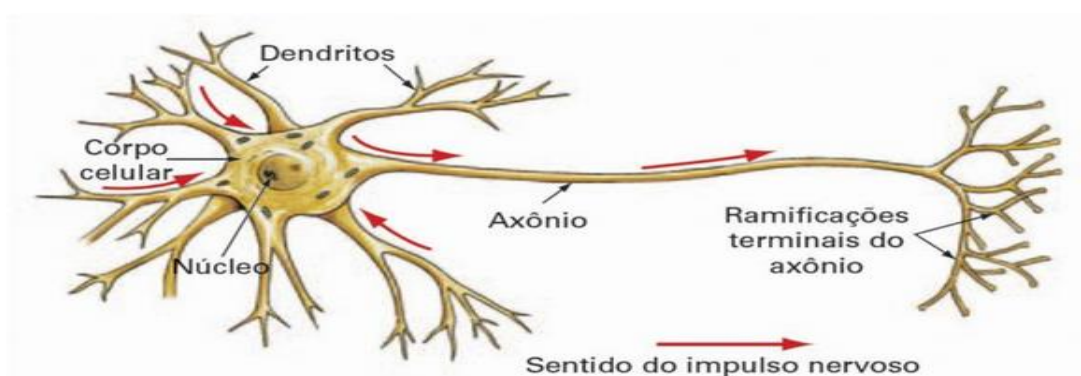
QUADRO 2 – Mapeamento e características das uninormas e nullnormas.

Normas Triangulares	
Uninorma (U)	Nullnorma (NU)
Mapeamento: $U: [0, 1] \times [0, 1] \rightarrow [0, 1]$	Mapeamento: $NU: [0, 1] \times [0, 1] \rightarrow [0, 1]$
Características	Características
(1) $U(x, y; g) = U(y, x; g)$ (Comutatividade)	(1) $NU(a, b) = NU(b, a)$ (Comutatividade)
(2) $U(x, y; g) \geq U(z, v; g)$ para $x > z$ e $y > v$ (Monotonicidade)	(2) $NU(a, b) \leq NU(a, c)$, se $b \leq c$ (Monotonicidade)
(3) $U(x, U(y, z; g); g) = U(U(x, y; g), z; g)$ (Associatividade)	(3) $NU(a, NU(b, c)) = NU(NU(a, b), c)$, (Associatividade)
(4) Elemento Neutro (Identidade): \exists algum $g \in [0, 1]$ tal que $U(x, g; g) = x \forall x \in [0, 1]$	(4) Elemento Neutro (Absorção): $NU(a, 0) = a \forall a \in [0, e]$ e $NU(a, 1) = a \forall a \in [e, 1]$, onde $a, b, c \in [0, 1]$.

Fonte: Adaptado de (Yager & Ribalov, 1996).

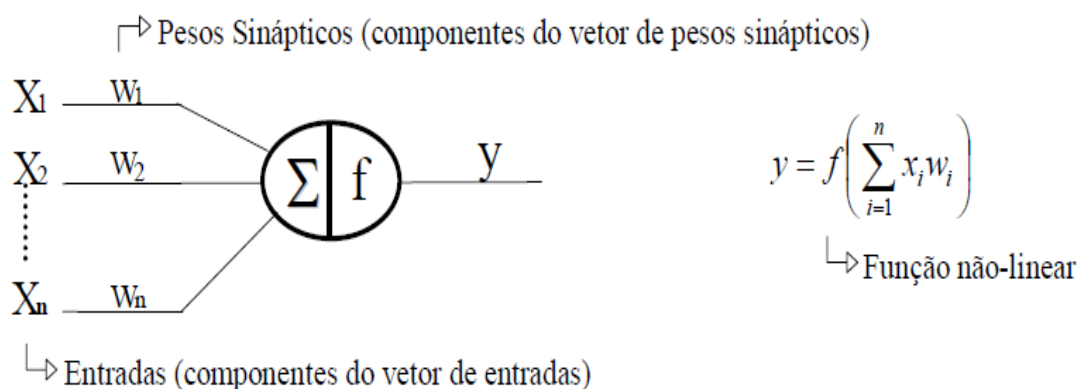
2.3 Neurônio Nebuloso Genérico

Os sistemas inteligentes buscam simular o comportamento humano em modelos para resolver problemas do cotidiano. Nos seres humanos o cérebro processa as informações de forma paralela, distribuindo as tarefas para as células nervosas. O cérebro é constituído de inúmeras células nervosas, onde os responsáveis pelas reações aos estímulos do ambiente são os neurônios. Eles são responsáveis em tratar os estímulos, ou seja, processar os sinais provenientes de uma cadeia de neurônios ou de situações externas. Em sua composição o neurônio apresenta dendritos, o axônio, o corpo celular e as sinapses. Os neurônios recebem os sinais de entrada provenientes de outras células através dos dendritos, processa-os no corpo celular e gera os sinais de saída que são transmitidos para outros neurônios por meio dos axônios e de suas ramificações (Calvo, 2007). A Figura 1 exemplifica e ilustra as partes de um neurônio biológico.



Fonte: Wasserman (1999)
Figura 1- Neurônio biológico.

Em 1943 baseados nos conceitos do neurônio, McCulloch e Pitts resolveram formular um modelo computacional baseado nas definições e características do neurônio biológico, ao qual deram o nome de neurônio artificial. Essa modelagem do neurônio desencadeou inúmeros estudos e pesquisas na área de redes neurais artificiais. O modelo apresentado em (McCulloch & Pitts, 1943) pode ser visualizado na Figura 2 (Calvo, 2007):



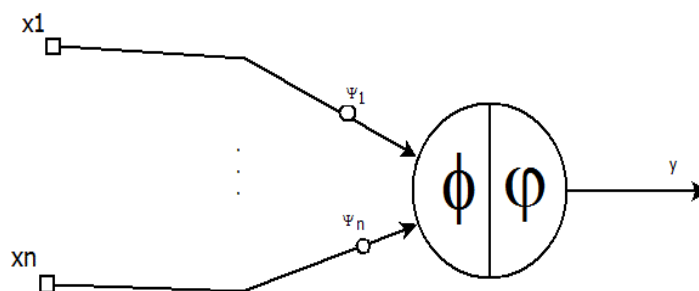
Fonte: Calvo (2007)
Figura 2- Neurônio artificial.

Dentre os vários estudos realizados para simular o comportamento do neurônio humano, destacamos os que buscaram agregar a natureza nebulosa ao modelo de neurônio artificial, buscando acrescentar ao modelo a habilidade de tratar informações imprecisas. A esse neurônio chamamos de neurônio nebuloso (Hell, 2008).

Um modelo geral de neurônio nebuloso é mostrado na Figura 3. Esse neurônio tem sua representação matemática descrita pela seguinte equação (2.1) (Ballini, Soares, & Andrade, 2003):

$$y(x) = \varphi(\phi(\Psi(x))) = \varphi(\phi(\Psi_1(x_1), \dots, \Psi_N(x_n))) \quad (2.1)$$

onde $\Psi: \mathfrak{R}^n \rightarrow \mathfrak{R}^n$ é o operador sináptico, $\phi: \mathfrak{R}^n \rightarrow \mathfrak{R}$ é o operador de agregação e $\varphi: \mathfrak{R} \rightarrow \mathfrak{R}$ é a função de ativação.



Fonte: Ballini, Soares, & Andrade (2003).
Figura 3- Neurônio nebuloso.

2.4 Classificação dos Neurônios Nebulosos

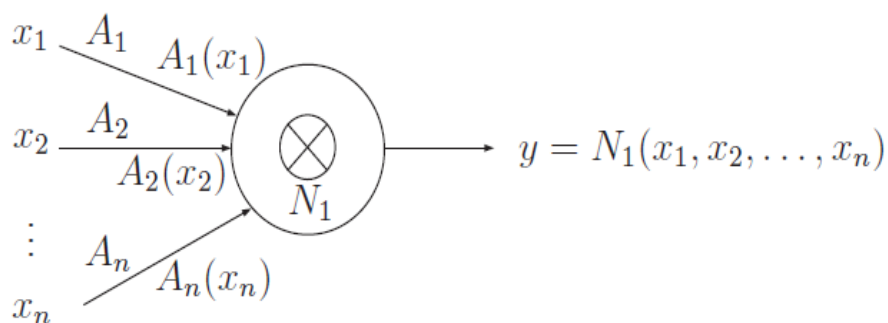
Inúmeros modelos de neurônios foram propostos, porém sua classificação divide-se em três tipos distintos, variando de acordo a utilização dos conceitos da lógica nebulosa na construção de sua estrutura (Lin & Lee, 1996 apud Hell, 2008):

- Neurônios nebulosos com entradas não nebulosas combinadas com pesos nebulosos (Tipo I).
- Neurônios nebulosos com entradas nebulosas que são combinados com pesos nebulosos (Tipo II).
- Neurônios Nebulosos descritos por equações lógicas nebulosas (Tipo III).

Os neurônios do tipo I são aqueles onde a entrada do modelo são números reais e os pesos de conexão são conjuntos nebulosos, chamados pesos nebulosos. Portanto, para n entradas não nebulosas x_1, x_2, \dots, x_n existem n conjuntos nebulosos A_i , $i = 1, \dots, n$, sendo que as operações de ponderação dos pesos são trocadas por operações com funções de pertinência, também conhecidas como operações de *fuzzificação*. O resultado de cada operação de *fuzzificação*, denotado por $A_i(x_i)$ é o valor de associação da entrada x_i no conjunto nebuloso A_i (Hell, 2008). O mapeamento de um neurônio do tipo I é $N_1 = \mathbb{R}^n \rightarrow [0, 1]$ definido da seguinte maneira:

$$N_1(x_1, x_2, \dots, x_n) = A_1(x_1) \otimes A_2(x_2) \otimes \dots \otimes A_n(x_n) \quad (2.2)$$

onde o operador de agregação \otimes pode ser qualquer mecanismo de agregação tais como os operadores min, max, ou qualquer t-norma ou s-norma (Hell, 2008). A Figura 4 ilustra um neurônio do tipo I.



Fonte: Hell (2008).

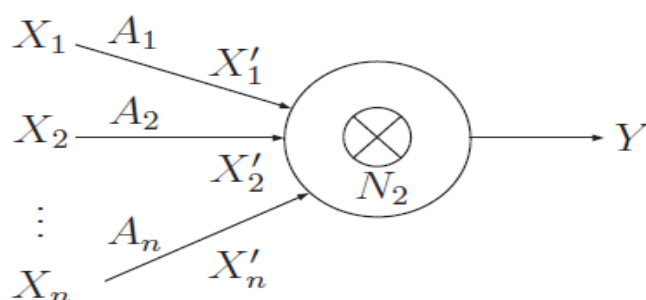
Figura 4- Neurônio nebuloso tipo I.

Já os neurônios do tipo II, chamados de N_2 são muito parecido aos neurônios de tipo I, exceto pelas entradas que se tratam de conjuntos nebulosos. Cada uma das entradas não nebulosas x_i é composta com seu respectivo peso nebuloso A_i através de um operador de ponderação (*). O resultado desta ponderação é outro conjunto nebuloso $X'_i = A_i * x_i$, $i = 1, \dots, n$. Todos os conjuntos nebulosos X'_i são então associados através de um operador de agregação \otimes para produzir o conjunto nebuloso de saída Y . O mapeamento do neurônio N_2 é $[0,1]^n \rightarrow [0,1]$ e pode ser descrito na forma matemática conforme a equação (2.3) a seguir (Hell, 2008):

$$X'_i = A_i * x_i, \quad i = 1, \dots, n.$$

$$Y = N_2(X'_1, X'_2, \dots, X'_n) = X'_1 \otimes X'_2 \otimes \dots \otimes X'_n \quad (2.3)$$

A Figura 5 revela a estrutura de um neurônio nebuloso do tipo II.



Fonte: Hell (2008).

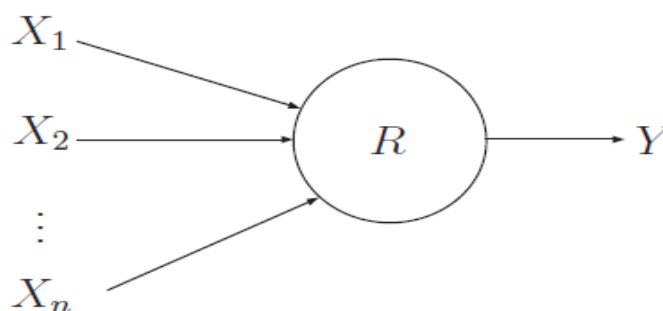
Figura 5- Neurônio nebuloso tipo II.

Por fim, os neurônios nebulosos denotados como tipo III são representados por regras nebulosas da forma se-então, de acordo com a equação (2.4) (Hell, 2008):

$$SE X_1 E X_2 \dots E X_n ENTÃO Y \quad (2.4)$$

Onde X_1, X_2, X_n são as entradas e Y é a saída do neurônio nebuloso.

Para este neurônio as entradas podem ser tanto nebulosas quanto não nebulosas, sendo que um neurônio nebuloso do tipo III com entradas não nebulosas pode ser visto como um caso especial deste neurônio com entradas nebulosas. Uma rede neural nebulosa composta por neurônios do tipo III pode ser vista como uma arquitetura para extração de regras a partir de dados de treinamento (Hell, 2008). A Figura 6 ilustra um exemplo de um neurônio do tipo III e R é uma relação nebulosa do tipo $X_1 \times X_2 \times X_n \times Y$.



Fonte: Hell (2008).

Figura 6- Neurônio nebuloso tipo III.

Os neurônios nebulosos do tipo III possuem algumas particularidades, onde podemos encontrar variações de modelos que combinam os neurônios nebulosos do tipo II e III. A esses neurônios chamamos de neurônios lógicos. Foram propostos por Pedrycz & Rocha em 1993 e apresentam uma estrutura similar aos neurônios do tipo III, excetuando o fato de que os pesos de conexão w_i serem associados a cada uma das entradas x_i . Esses neurônios executam um mapeamento no espaço formado pelo produto cartesiano entre o espaço de entrada e o espaço dos pesos no intervalo unitário, isto é $X \times W \rightarrow [0,1]$ (Hell, 2008). Exemplos de tais neurônios são os neurônios *and* e *or* (Pedrycz & Rocha, 1993) e o *unineuron* (Pedrycz, 1993), (Lemos et al., 2012).

O neurônio lógico *and* realiza uma ponderação das entradas por pesos através de um operador s-norma e gera a saída agregando os resultados da operação de

ponderação por uma t-norma. O neurônio lógico *or* utiliza uma t-norma na operação de ponderação e uma s-norma na agregação final (Hell, 2008). O *unineuron* é visto como uma extensão dos neurônios lógicos *and* e *or* (Pedrycz, 1993) em que uma uninorma (Yager & Rybalov, 1996), que é uma generalização das t-normas e s-normas, é utilizada nas operações de ponderação e agregação (Pedrycz, 1993), ou apenas na operação de agregação (Lemos et al, 2012). A uninorma é um mapeamento $U: [0,1]^n \rightarrow [0,1]$ que possui como elemento identidade $g \in [0,1]$ tal que $U(x,g)=x$. Através desse elemento identidade as uninormas estendem t-normas e s-normas através de variação do valor g no intervalo entre 0 e 1, permitindo a alternância entre uma s-norma ($g=0$) e t-norma ($g=1$) (Lemos et al, 2010). Um exemplo de uninorma a ser utilizada por *unineurons* foi formulado por Lemos et al (2010) e é expressa da seguinte maneira:

$$U(x, y) = \begin{cases} gT\left(\frac{x}{g}, \frac{y}{g}\right) & , se x, y \in [0, g] \\ g + (1 - g)S\left(\frac{x-g}{1-g}, \frac{y-g}{1-g}\right) & , se x, y \in [g, 1] \\ \max(x, y) \text{ ou } \min(x, y) & , caso ontrário \end{cases} \quad (2.5)$$

onde T é uma t-norma e S é uma s-norma.

Dado um vetor de entrada $\mathbf{x} [x_1, x_2, \dots, x_n]$ e um vetor de pesos do neurônio $\mathbf{w} [w_1, w_2, \dots, w_n]$ para $x_i \in [0, 1]$ e $w_i \in [0, 1]$ para i de $1, \dots, n$. A saída dos neurônios lógicos *and* e *or* para essa situação específica são descritas como (Pedrycz, 1993):

$$z = OR(x, w) = S_{i=1}^n(x_i t w_i) \quad (2.6)$$

$$z = AND(x, w) = T_{i=1}^n(x_i s w_i) \quad (2.7)$$

onde T e t são t-normas e S e s s-normas.

O *unineuron* proposto em Lemos et al (2012) realiza as seguintes operações para computar sua saída:

- transforma-se cada par (x_i, w_i) em um único valor $b_i = h(x_i, w_i)$;
- calcula-se a agregação uninorma dos valores transformados $U(b_1, b_2, \dots, b_n)$, onde n é o número de entradas.

A função h é responsável por transformar as entradas e os pesos correspondentes em valores transformados individuais (Lemos et al, 2012). Uma formulação para a função h pode ser descrita como (Yager & Rybalov, 1996):

$$h(w, x) = wx + wg \quad (2.8)$$

Utilizando a agregação ponderada relatada em (2.8) podemos escrever o *unineuron* como (Lemos et al, 2010):

$$z = UNI(w; x; g) = U_{i=1}^n h(w_i x_i) \quad (2.9)$$

O diagrama a seguir (Figura 7) sintetiza como os vários tipos de neurônios e suas classificações estão dispostas.

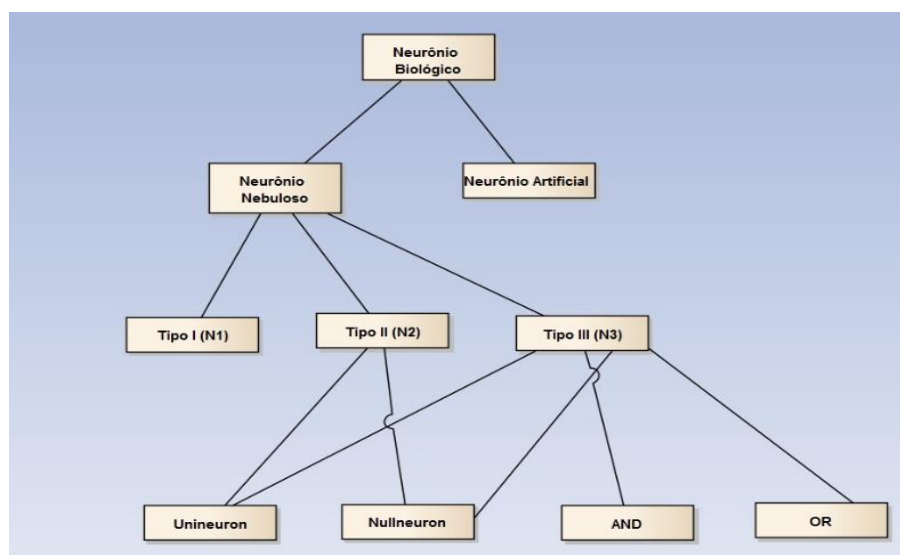


Figura 7- Diagramas de neurônios e suas especializações.

2.5 Redes Neurais Nebulosas

As redes neurais nebulosas utilizam-se da estrutura de uma rede neural artificial, onde os neurônios artificiais clássicos são substituídos por neurônios nebulosos (Pedrycz, 1993 apud Hell, 2008). Estes neurônios são implementados por meios de normas triangulares, que generalizam as operações de união e interseção de conjuntos clássicos para a teoria de conjuntos nebulosos. Assim, a rede neural passa a ser vista como um sistema linguístico, preservando a capacidade de aprendizado da RNA (Hell, 2008).

A utilização de neurônios nebulosos acarreta em uma rede com topologia interpretável. A principal característica dessas redes é a sua transparência, permitindo a utilização de uma informação a priori para definir a topologia da rede inicial e permitindo a extração de informações valiosas a partir da topologia resultante após o treinamento na forma de um conjunto de regras *fuzzy* (Hell, 2008).

A seguir podemos visualizar algumas características de redes neurais nebulosas formuladas, evidenciando alguns de seus aspectos importantes, como por exemplo, quantidade de camadas em sua arquitetura, metodologias de agrupamento de granularização de dados, métodos de treinamento, dentre outros.

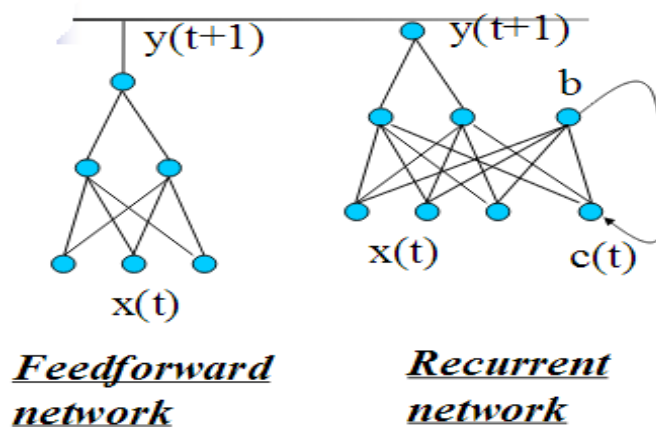
2.6 Características de Redes Neurais Nebulosas

As redes neurais nebulosas possuem várias características das quais podemos distinguir os modelos através de suas propriedades básicas, como a forma de conexão da rede, o tipo de neurônio nebuloso utilizado, o tipo de aprendizado (treinamento) e a forma como as entradas são tratadas nas primeiras camadas dos modelos. A seguir são esclarecidas algumas dessas características importantes dos modelos de redes neurais nebulosas.

As redes neurais nebulosas podem ser classificadas em relação a como seus neurônios estão conectados. Essa forma de conexão define como os sinais serão transmitidos na rede. Em geral existe a alimentação à frente (*feedforward*) onde os neurônios nebulosos são agrupados em camadas e o sinal percorre toda a rede em um único sentido, geralmente da entrada do modelo até a sua saída gerando um resultado esperado. Os neurônios *fuzzy* na mesma camada não possuem conexão e suas redes também são conhecidas como redes sem realimentação (Haykin, 2001). Esse tipo de conexão é a mais comum entre os modelos de redes neurais nebulosas, onde podemos citar os modelos desenvolvidos por Lemos et al (2010, 2012), Bordignon e Gomide (2012) e Silva, Caminhas, Lemos & Gomide (2012), dentre outros.

Por fim existem as redes que são utilizadas com realimentação, também chamadas de recorrentes (*recurrent*). Nesse tipo de rede os neurônios nebulosos também são reunidos em camadas, porém existe alimentação de informações em neurônios na mesma camada, podendo inclusive acontecer com o próprio neurônio nebuloso em questão, ou até mesmo em camadas anteriores, caso existam. Nesse tipo de rede o sinal percorre a rede em duas direções, diferente das redes *feedforward* e tem capacidade de representar estados em sistemas dinâmicos (Haykin, 2001). Uma rede neural nebulosa que possui esse

tipo de conexão pode ser visualizada no modelo desenvolvido por Luna, Ballini, & Gomide (2004). A Figura 8 exemplifica as duas formas de conexões entre os neurônios explicados.



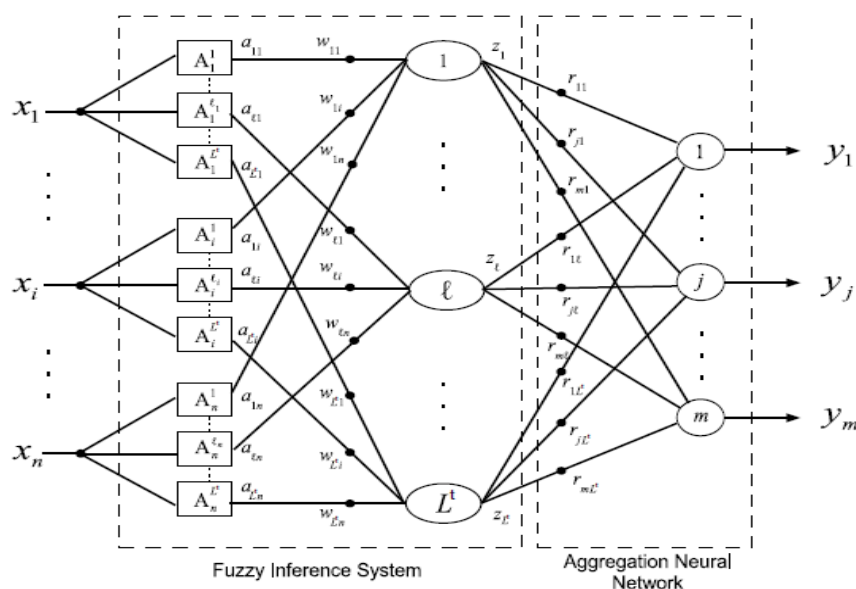
Fonte: <http://slideplayer.us/slide/217113/> (acessado em agosto de 2014).

Figura 8- Exemplo de redes *feedforward* e redes *recurrent*.

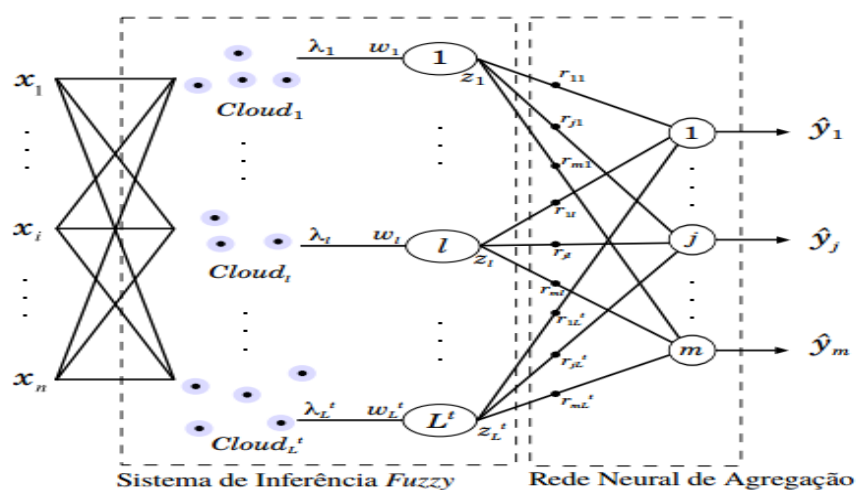
A quantidade de camadas também é um fator que pode diferenciar as redes neurais nebulosas. Nos modelos, cada camada é responsável por uma função ou tarefa específica. Em geral a primeira camada é responsável por tratar as entradas e a última por trazer a resposta da rede. Entre essas duas camadas existem outras intermediárias, que podem ser ocultas ou não. Dependendo do modelo e ao que este se propõe, cada camada tem uma função específica. Das redes neurais artificiais que possuem três camadas, alguns modelos possuem duas camadas baseadas em conceitos de sistemas nebulosos e a terceira é uma rede neural artificial. Nesses modelos, a primeira camada é responsável por receber as entradas e possui neurônios onde a função de ativação é uma função de pertinência de conjuntos nebulosos definidos pela granularização do espaço de entrada. Os modelos de Lemos et al (2010, 2012), Bordignon & Gomide (2012) são exemplos de modelos que apresentam essas características. A segunda camada pode apresentar neurônios nebulosos que realizam agregação dos valores oriundos da camada anterior, onde o tipo de neurônio pode variar conforme a topologia e/ou o modelo escolhido. Os modelos de Hell et al (2008) e Lemos et al (2010) utilizam neurônios nebulosos (unineurons) para agregar os valores da primeira camada e em contrapartida o modelo proposto por Luna, Ballini & Gomide (2004), utiliza neurônios AND que podem ser utilizados de forma recorrente. Por fim a última camada na maioria das vezes é uma rede neural que é representada por um neurônio clássico desempenhando uma função de agregação. O modelo proposto por Rosa & Gomide, (2013) utiliza uma rede neural com funções sigmodais

na última camada, assim como o modelo de Bordignon & Gomide (2012^b). Suas estruturas podem ser visualizadas nas Figuras 9(b) e 9(a) respectivamente.

Nos modelos que possuem quatro e cinco camadas em sua arquitetura destacamos as redes formuladas por Porwal, Carranza e Hale (2004), Leite, Costa e Gomide (2010) e Kim & Kasabov (1999). Além das três camadas destacadas anteriormente o modelo de Powell et al (2004) possui uma camada para realizar regras nebulosas do tipo se-então. Na Figura 10 podemos identificar a estrutura do modelo com destaque para suas camadas. Já o modelo *neuro-fuzzy* proposto no Kim et al (1999) é uma rede neural multicamadas baseadas em sistemas fuzzy. Nesta estrutura conectiva, os nós de entrada e de saída representam os estados de entrada e sinais de controle de saída/decisão respectivamente, e nas camadas escondidas, existem gânglios que funcionam como funções de pertinência (MFS) e regras. Isso elimina a desvantagem de uma rede *feedforward* multicamada normal, que é difícil para um observador de compreender ou modificar (Kim & Kasabov 1999) O modelo proposto encontra-se ilustrado na Figura 11, onde estão destacadas as cinco camadas da rede neural nebulosa. Sobre a quantidade de camadas ainda podemos destacar modelos que possuem uma quantidade indeterminada de camadas, que são definidas conforme os parâmetros e valores envolvidos no sistema. O modelo proposto por Mitrakis, Theocharis & Petridis (2008) tem esse comportamento.



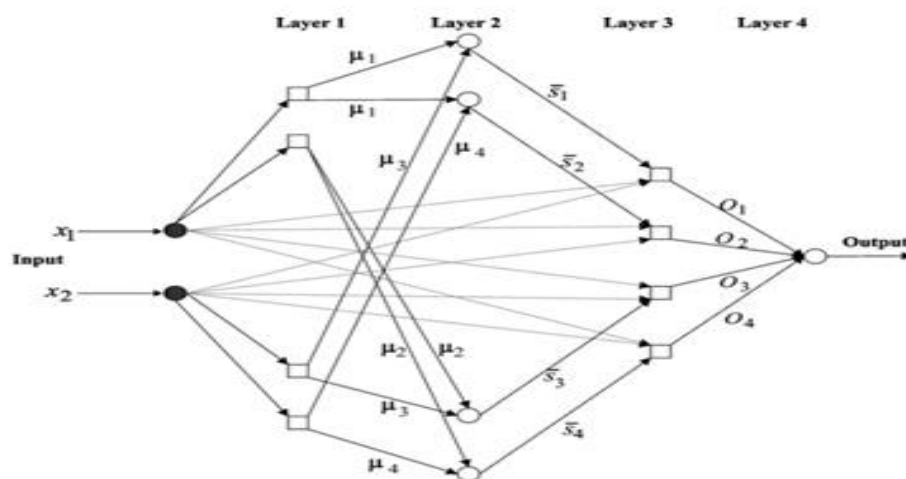
(a)



(b)

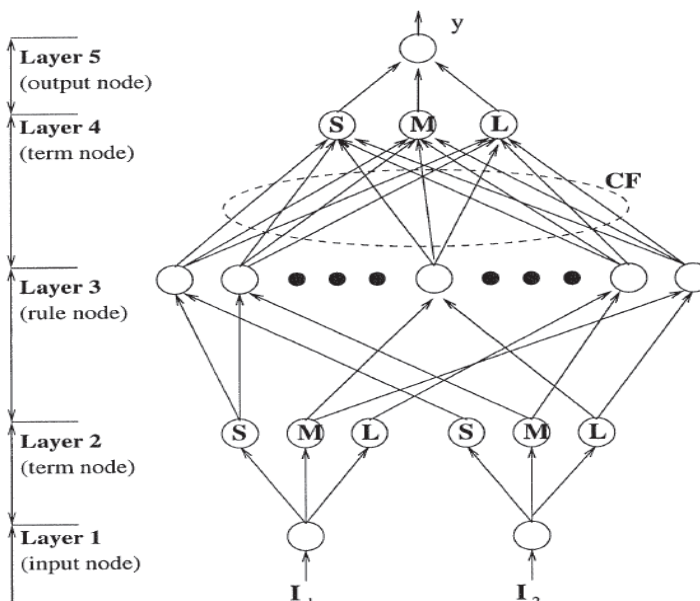
Fonte: Bordignon & Gomide (2012) e Rosa et al (2013).

Figura 9- Topologias de redes neurais nebulosas com três camadas.



Fonte: Porwal et al (2004).

Figura 10- Arquitetura simplificada do modelo ANFIS com quatro camadas.



Fonte: Kim & Kasabov (1999).

Figura 11- Estrutura da rede neural HyFIS com cinco camadas.

Quando avaliamos as redes sobre os tipos de neurônios que compõem suas camadas, podemos destacar os modelos que possuem os neurônios nebulosos do tipo AND/OR.

Esses modelos em geral utilizam essas estruturas para usar operadores de t-norma e s-norma. Como exemplo de redes neurais nebulosas que utilizam esses tipos de neurônio destacamos os modelos Caminhas, Tavares, Gomide & Pedrycz (1999) de Porwal, Kym e Kasabov (1999) e Carranza & Hale (2004). Já o modelo de Luna, Ballini, & Gomide (2004) utilizam o neurônio lógico AND de forma recorrente devido à estrutura e metodologia empregada em seu modelo.

Dos tipos de neurônios nebulosos muito utilizados nas redes neurais nebulosas destacamos aqueles baseados em nullnormas e uninormas, chamados respectivamente de *nullneuron* e *unineurons*. Modelos como o de Lemos, Caminhas & Gomide (2010, 2012), Leite Costa & Gomide (2010) Bordignon & Gomide (2012, 2012^b) exemplificam redes neurais nebulosas baseadas nesses neurônios.

Já o modelo de Silva, Caminhas, Lemos & Gomide (2012) é construído utilizando uma estrutura de rede neural nebulosa com neurônios do tipo *Neo-Fuzzy-Neurons*. Basicamente, a estrutura da rede NFN (*Neo-Fuzzy-Neurons*) utilizada nesse modelo corresponde a um conjunto de n regras *fuzzy* (Silva et al, 2012).

Outros modelos de neurônios também podem ser utilizados, como por exemplo, o classificador de neurônio nebuloso, proposto por Mitrakis, Theocharis & Petridis (2008).

Avaliando o tipo de treinamento para as redes neurais nebulosas podemos destacar que esses algoritmos são um conjunto de passos bem definidas para a resolução de problemas de aprendizado. Essas metodologias de treinamento buscam simular o aprendizado humano ao aprender ou atualizar seus novos conceitos, atuando principalmente na atualização de fatores da rede, como por exemplo, os valores dos pesos sinápticos. Em geral, tanto para as redes neurais artificiais quanto para as redes neurais nebulosas o aprendizado pode ser classificado conforme os seguintes paradigmas:

- a) Supervisionado- quando é utilizado algum fator externo que indica à rede qual é o resultado desejado para o problema em questão.
- b) Não supervisionado - quando não existe nenhum agente externo indicando a resposta da rede conforme os padrões de entrada submetidos aos modelos.
- c) Reforço - quando um avaliador externo afere a resposta do modelo (Haykin, 2001).

Um modelo empregado para treinamento de rede e atualização de elementos envolvidos na rede são as máquinas de aprendizado extremo (ELM) (Huang et al, 2006), que tem se mostrado um método rápido e bastante preciso para o ajuste de parâmetros, sendo utilizado inclusive por vários modelos de redes distintos (Lemos et al, 2012, Bordignon e Gomide (2012, 2012^b), Rosa & Gomide (2013)).

Além desses conceitos, podem ser utilizados algoritmos genéticos para realizar o treinamento da rede. O modelo proposto por Lemos et al (2010) utiliza um algoritmo genético chamado *real coded genetic algorithm* (Herrera e Verdegay, 1998), encontrando os parâmetros do modelo que minimizam a medida do erro quadrado médio utilizando o conjunto de dados de treinamento para atualizar variáveis importantes da rede. Cada indivíduo gerado pelo algoritmo genético codifica os seguintes atributos da rede: dispersão das funções de pertinência gaussianas dos conjuntos nebulosos; pesos que ligam a primeira e segunda camada; elementos de identidade dos *unineurons* da segunda camada; *singletons* nebulosos (Lemos et al, 2010).

O método de aprendizagem do modelo Luna, Ballini, & Gomide (2004) basearam-se em dois procedimentos: o método do gradiente da função erro, para os pesos

da camada de saída e o método de aprendizado por reforço associativo, para os pesos da camada intermediária.

Para realizar a aprendizagem em seu modelo Silva, Caminhas, Lemos & Gomide (2012) propuseram uma adaptação dos pesos da NFN usando um método baseado no gradiente com taxa de aprendizado ótima. O aprendizado é supervisionado e tem o propósito de ajustar os valores dos parâmetros. O procedimento de adaptação dos pesos é um mecanismo incremental de atualização, portanto apropriado para modelos evolutivos (Silva et al,2012).

Na rede neural nebulosa de Porwal, Carranza, & Hale (2004) o treinamento envolve ajuste iterativo de parâmetros do sistema usando um procedimento de aprendizagem híbrida para mapear cada vetor de treinamento para seu vetor alvo de saída com valor mínimo de erro quadrático. O sistema de inferência *neuro-fuzzy* adaptativo treinado é usado para processar todos os vetores de características.

Já o trabalho proposto por Leite et al (2010) sugere uma abordagem de rede neural granular (eGNN) para desenvolver modelos de classificação online, sendo que o algoritmo de aprendizado incremental se adapta a estrutura eGNN e seus parâmetros sempre que existam alterações do sistema. O algoritmo pode lidar com entradas marcadas e não marcadas concomitantemente.

Por fim o HyFIS (*Hybrid neural System Inference fuzzy*), desenvolve um esquema de aprendizagem de duas fases. Na primeira fase, realizada no módulo de aquisição de conhecimento, um método para derivar regras *fuzzy* de pares de dados de entrada e saída desejados, proposto por Wang & Medal (1992), é usado para encontrar regras de lógica nebulosa e adequada à estrutura inicial do sistema *neuro-fuzzy*. Na segunda fase, uma técnica de aprendizagem de parâmetro, usando um algoritmo aprendizagem baseado em gradiente descendente, é aplicada a funções de pertinência de entrada/saída de variáveis linguísticas.

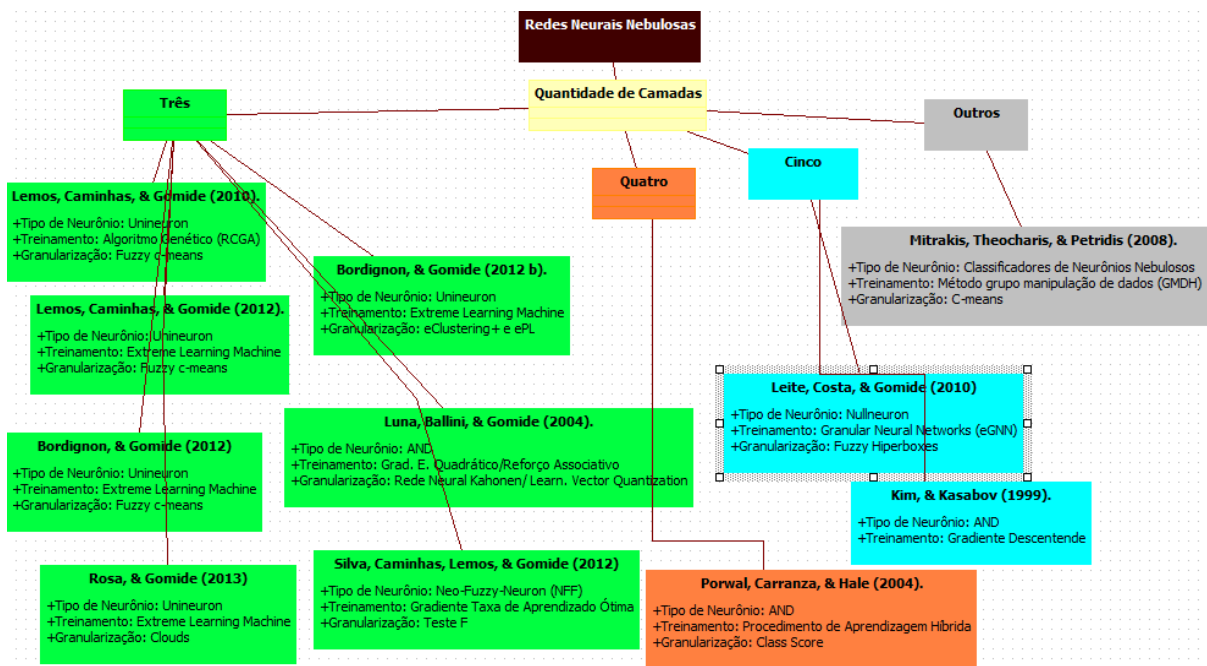
A maneira como as entradas são granularizadas nas redes neurais nebulosas também possuem destaque, pois a forma com que elas atuam reflete diretamente nas variáveis envolvidas no modelo. O método com que as entradas são manipuladas podem definir parâmetros como funções de ativação, funções de pertinência de conjuntos nebulosos, topologia de rede, dentre outros (Lemos et al, 2010). Redes neurais nebulosas

podem utilizar métodos de agrupamento, como o *c-means* e sua versão nebulosa o *fuzzy c-means* (Dunn, 1973 e Bezdec, 1981), métodos baseados em densidade dos dados (*clouds* (Angelov & Yager, 2012), agrupamento online (*eClustering+* (Angelov, 2010) e o ePL (Lima, Gomide & Ballini, 2006), testes estatísticos (teste F) (Alelen, 1997) , dentre outros. A seguir algumas definições e exemplos de utilização.

O c-Means proposto por Macqueen em 1967 foi um dos algoritmos pioneiros buscando realizar a análise de agrupamento. Nesse algoritmo cada vetor protótipo sempre está associado à representação de uma classe ou grupo do conjunto de dados e, para isso, deve residir no mesmo espaço \mathbb{R}^n que os dados do conjunto. O modelo proposto por Mitrakis, Theocharis & Petridis (2008) utiliza essa metodologia para tratar as suas entradas.

O *Fuzzy c-means* é um método de agrupamento que foi desenvolvido por Dunn em 1973 e aprimorado por Bezdec em 1981 permitindo que um conjunto de dados pertença a dois um mais grupos baseado em suas características de similaridade em relação ao grupo analisado, chamados de *clusters* (Bezdec, 1981). Seu funcionamento é muito parecido com o método c-means, excetuando pela utilização de graus de pertinência da informação avaliada em relação ao grupo alvo, assim como na lógica nebulosa. Essa característica possibilita que dados que estão na fronteira entre dois clusters possuam graus de pertinência adequados a cada uma delas (Nock & Nielsen, 2006). É uma escolha comum para granularizar o espaço de entrada em diversos trabalhos (Lemos; Caminhas & Gomide, 2010, 2012, Bordignon & Gomide, 2012). O algoritmo fornece a distância Euclidiana até o centro do grupo mais próximo para ser utilizada como a dispersão σ dos grupos. Os centros dos grupos são projetados nos eixos correspondentes e são formados conjuntos fuzzy Gaussianos, centrados nas projeções dos respectivos valores modais (centros) e com dispersão σ (Bordignon et al, 2012);

Os *Clouds*, propostos por Angelov e Yager (2012) tem o propósito de criar uma partição no espaço de entrada sem formar de forma explícita o agrupamento, como realiza o *fuzzy c-means*. A *Cloud* é caracterizada pela densidade que representa uma distribuição do espaço de entrada. A coleção dos *Clouds* faz o papel análogo ao de grupos formados pelo *fuzzy c-means*, granularizando o espaço de entrada. O modelo proposto em (Rosa et al, 2013) utiliza *Clouds* para tratar o espaço de entrada segundo a densidade dos dados de forma recursiva. Com os *Clouds* são geradas as funções de pertinência gaussianas. A Figura 12 traz um resumo de alguns algoritmos classificados conforme a quantidade de camadas.



Fonte: Dados da própria pesquisa.

Figura 12- Diagramas de modelos de redes neurais nebulosas diversas e suas especificidades.

2.7 Métodos de Regularização

Quando tratamos de modelos, alguns fatores podem prejudicar a busca pela resposta ideal. Um dos que podem prejudicar a resposta de um modelo é a incerteza. Em modelos como redes neurais nebulosas, os seguintes fatores são problemas comuns que podem gerar imprecisões nas respostas finais das estimativas de comportamento ou resultado a um dado evento:

- dados insuficientes;
- informações em excesso;
- operações insuficientes.

No terceiro fator, podemos ter dados suficientes para resolver as ambiguidades, porém os problemas podem não gerar o resultado esperado devido a um treinamento incorreto, aproximações ou falhas (Battiti, 1994).

Procedimentos que exploram restrições para suavizar problemas de aproximação com base em informações de entrada/saída do modelo são conhecidos como regularização, tendo sido inseridas na literatura por Tikhonov em 1963 (Tikhonov, 1963). A

regularização é a metodologia de aquisição de um problema de aproximação bem-comportado através de um problema mal posto, realizado pela incorporação de restrições de suavidade junto ao modelo de aproximação. Esse processo conduz a obtenção de modelos de aproximação bem estabelecidos, pois permite a definição de um processo contínuo e automático, não necessariamente singular, que produz a “melhor aproximação” baseada nos dados de entrada/saída da rede.

O processo de regularização tende para um problema de otimização que tem como principal foco minimizar uma função objetivo que apresenta dois termos básicos: um que expressa o erro de aproximação e o segundo expressa o nível de distanciamento da condição de suavidade. Quando existe a regularização da função que será aproximada, isso impede que o comportamento da função seja inteiramente local, permitindo o processo de generalização (Tikhonov, 1963).

A seguir destacaremos três métodos de regularização muito utilizados: regressão *ridge*, *lasso* e *elastic net*.

2.8 Regressão *Ridge*

A regressão *ridge* é um método que foi proposto por Hoerl & Kennard em 1970 que trabalha com metodologias de regressão baseadas na penalização da soma de quadrados. A regressão *ridge* (também chamada de regressão de *Tikhonov* ou regressão de *cume*) encolhe os coeficientes de regressão através de uma imposição de penalidade a seu tamanho. Os coeficientes *ridge* minimizam a soma de quadrados residual penalizado. A forma que representa essa minimização segue abaixo (Hastie et al, 2009):

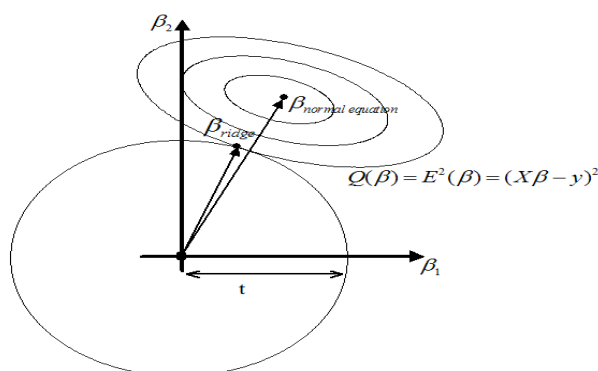
$$\beta_{ridge} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p \beta_j^2 \right\} \quad (2.10)$$

Onde n é igual ao número de amostras, β_j são parâmetros ou coeficientes desconhecidos, p é o número de colunas das amostras, x_i e y_i são respectivamente a i -ésima entrada e saída dos dados de treinamento $\lambda \geq 0$ é um parâmetro de complexidade que controla a quantidade de encolhimento: quanto maior o valor de λ , maior será a diminuição. (Hastie et al, 2009).

A ideia de penalização pela soma dos quadrados dos parâmetros também é usado em redes neurais, onde é conhecido como decaimento de peso, conforme explica Hastie (2009) . Uma maneira equivalente de escrever o problema *ridge* é:

$$\begin{aligned} \beta_{ridge} &= \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2, \\ \text{sujeito a } \sum_{j=1}^p \beta_j^2 &\leq t \end{aligned} \quad (2.11)$$

O que torna explícita a restrição de tamanho nos parâmetros. Existe uma correspondência de um para um entre os parâmetros λ em (2.10) e t em (2.11). Quando há muitas variáveis correlacionadas em um modelo de regressão linear, os seus coeficientes podem tornar-se mal determinados e apresentar alta variância. Um grande coeficiente positivo de uma variável pode ser cancelado por um coeficiente negativo igualmente grande sobre seu par correlacionado. Ao impor uma restrição de tamanho para os coeficientes, como em (2.11), este problema é aliviado (Hastie et al, 2009). A Figura 13 ilustra a técnica de regressão *ridge* que é na verdade a busca por um valor β_* (β_{ridge} na figura) que minimiza o erro quadrático de $(\beta) = (X\beta - y)^2$ sobre a limitação de que $\|\beta\|_2 < t$.

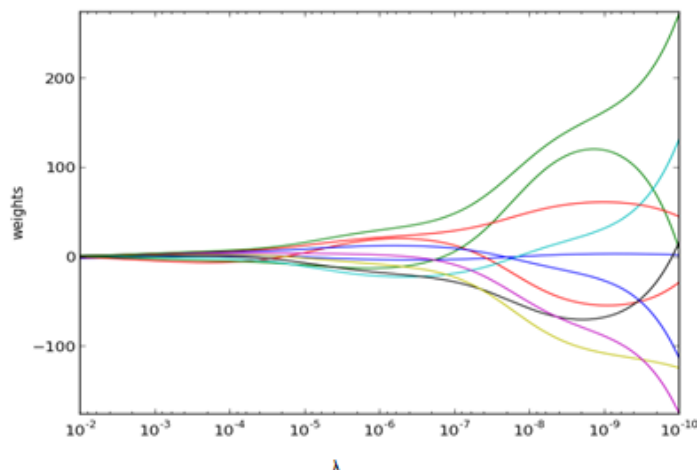


Fonte: Adaptado de <http://www.applied-mathematics.net/identification/lars.html>

Figura 13-Regressão Ridge

A Figura 14 apresenta os valores estimados para os coeficientes de um dado modelo em função do decréscimo de λ . Cada cor do gráfico representa um elemento distinto do vetor de coeficientes. No final da figura visualizamos coeficientes com grandes oscilações devido à disposição do valor de λ ¹ tender a zero e a solução tender aos mínimos quadrados.

¹ λ é um parâmetro de complexidade que controla o valor de encolhimento dos coeficientes. Quanto maior o valor de λ , maior será a quantidade de encolhimento tornando assim os coeficientes mais robustos para a colinearidade.



Fonte: Adaptado de http://scikit-learn.org/stable/auto_examples/linear_model/plot_ridge_path.html#example-linear-model-plot-ridge-path-py

Figura 14-Coefficientes *Ridge* como uma função de diminuição do parâmetro λ .

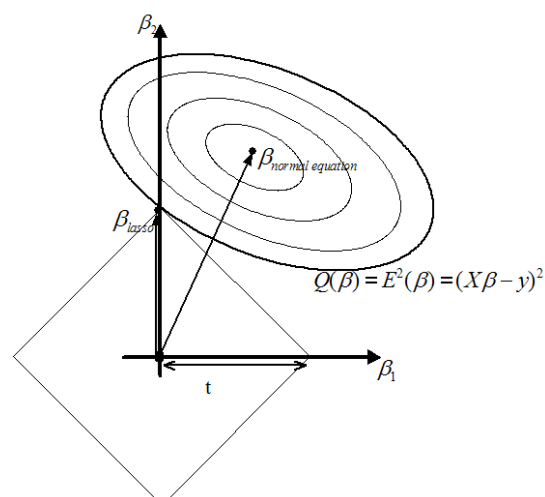
2.9 Método *lasso*

Conforme Hastie (2009) o método *lasso* é um processo similar à regressão *ridge*, porém com algumas peculiaridades na restrição da função de otimização, conforme demonstradas na equação a seguir:

$$\beta_{lasso} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2,$$

$$\text{sujeito a } \sum_{j=1}^p |\beta_j| \leq t \quad (2.12)$$

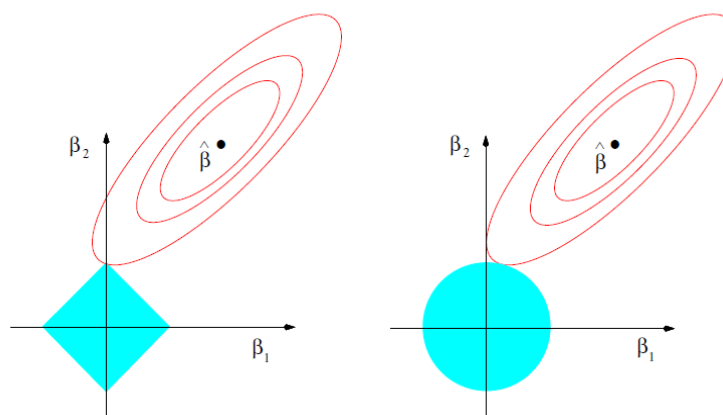
Comparando com a regressão *ridge*, podemos afirmar que esta substitui o termo de penalização $\sum_1^p \beta_j^2$ por $\sum_1^p |\beta_j|$. Essa restrição torna a solução não linear em função do y_i , não sendo possível definir uma expressão fechada para o cálculo dos coeficientes como na regressão *ridge* (Hastie et al, 2009). Na Figura 15 o método *lasso* busca o vetor β_* (β_{lasso} na figura) que minimiza o erro quadrático $(\beta) = (X\beta - y)^2$, porém com uma limitação diferente daquela utilizada pelo método *ridge*.



Fonte: Adaptado de <http://www.applied-mathematics.net/identification/lars.html>

Figura 15- Regressão lasso.

Já na Figura 16 abaixo, podemos comparar as estimativas do método *lasso* e *ridge*. Nessa figura são visualizados os contornos das funções de erro e suas restrições. As formas geométricas representam as áreas de restrições e as curvas de níveis representam as funções de erro de mínimos quadrados.



Fonte: Hastie et al (2009).

Figura 16- Imagem Estimativa para o *lasso* (esquerda) e regressão de *ridge* (direita). São mostrados os contornos das funções de erro e de restrição.

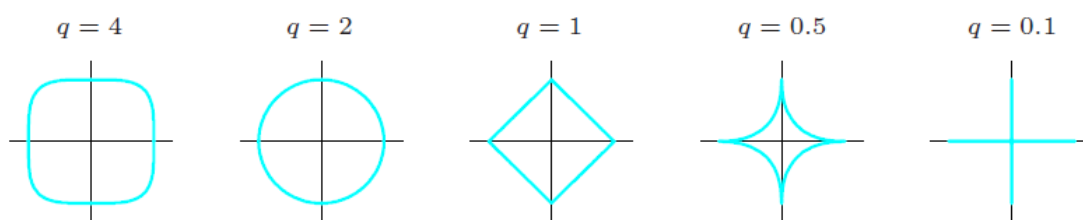
Na imagem evidenciada na Figura 16 identificamos o disco formado pela restrição *ridge* que representa $\beta_1^2 + \beta_2^2 \leq t$ e o diamante formado pela restrição *lasso* que é formada através da expressão $|\beta_1| + |\beta_2| \leq t$. Ambos os métodos mostrados buscam encontrar onde os contornos elípticos, que representam as funções de erro, encontram primeiramente a região de restrição. Ao avaliarmos as duas situações acima podemos

definir que, ao contrário do disco, o diamante é composto por cantos. Portanto, se as curvas de erro encontram primeiramente essa extremidade, por definição, estas possuem um parâmetro β_j com o valor zero. Quando o valor de $p > 2$ nas equações (2.10), (2.11) e (2.12) as definições de restrição tem sua forma geométrica alterada tornando-se rombóide, permitindo a figura que representa a restrição possua diversos cantos, bordas planas e faces, aumentando as possibilidades dos parâmetros estimados terem valores zero. Buscando a generalização da regressão *ridge* e *lasso* encontramos a seguinte equação (Hastie et al, 2009):

$$\boldsymbol{\beta} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \{ \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p x_{ij} \beta_j)^2 + \lambda \sum_{j=1}^p |\beta_j|^q \} \quad (2.13)$$

para $q \geq 0$, onde q é um parâmetro que define o formato das regiões de restrição.

Os contornos dos valores da constante $\sum_j |\beta_j|^q$ são mostradas na Figura 17 a seguir, para casos que o número de entradas é igual a 2. As regiões em azul são as figuras geométricas que resultam conforme a variação de q . Nessa figura podemos visualizar que a quantidade de cantos e faces varia à medida que o valor de q se altera, modificando também as possibilidades do encontro dos contornos elípticos de nível de erro com a região de restrição.



Fonte: Hastie et al (2009).

Figura 17- Contornos dos valores da constante $\sum_j |\beta_j|^q$ conforme o valor de q se altera.

Na expressão $\sum_j |\beta_j|^q$ a variação de q define qual tipo de método é utilizado para realizar a regularização. Quando $q=1$ trabalha-se com *lasso* e da mesma forma utiliza-se a regressão *ridge* quando $q = 2$. Nota-se ainda que à medida que o valor de q diminui os contornos que definem a região de restrição tendem a se aproximar das coordenadas. Em particular, quando o valor de $q = 1$ se obtém o menor valor de q onde a região de restrição é convexa. Quando uma região tem esse comportamento os problemas de otimização tornam-

se mais simples, diferentemente de quando a região de restrição tem características não convexas (Hastie et al, 2009).

2.9.1 Bolasso

Um algoritmo muito utilizado para estimar os parâmetros de um modelo de regressão e selecionar características relevantes é o *Least Angle Regression* (LARS) (Efron et al, 2004). O LARS é um algoritmo de regressão para dados com altas dimensões que é capaz de estimar não somente os coeficientes de regressão, mas também um subconjunto de regressores candidatos a serem incluídos no modelo final. Uma modificação ao algoritmo LARS implementa o *lasso* utilizando os mínimos quadrados ordinários que restringe a soma dos coeficientes de regressão (Efron et al, 2004). Ao avaliarmos um conjunto de n amostras distintas (x_i, y_i) , onde $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{iN}] \in \mathbb{R}^N$ e $y_i \in \mathbb{R}$ para $i=1, \dots, N$, a função de custo do algoritmo *lasso* pode ser definida como:

$$\sum_{i=1}^N \|\mathbf{z}(x_i)\mathbf{v} - y_i\|_{.2} + \lambda \|\mathbf{x}\| \quad (2.14)$$

onde λ é um parâmetro de regularização, comumente estimado via validação cruzada (Murphy, 2012). O primeiro termo de (2.14) corresponde à soma dos quadrados dos resíduos (RSS). Esse termo diminui à medida que o erro de treinamento decresce. O segundo termo é um termo de regularização L_1 . Geralmente esse termo é acrescentado, pois melhora a generalização do modelo, evitando o superajuste e é capaz de gerar modelos esparsos (Murphy, 2012).

O algoritmo LARS pode ser utilizado para efetuar a seleção do modelo, uma vez que para um dado valor de λ apenas uma fração (ou nenhuma) dos regressores tem pesos correspondentes diferentes de zero. Se $\lambda = 0$, o problema torna-se regressão irrestrita, e todos os pesos são diferentes de zero. À medida que aumenta λ , de 0 a um determinado valor λ_{max} , o número de pesos diferentes de zero diminui até chegar em 0. Para o problema considerado neste trabalho, os regressores z_{ls} são as saídas dos neurônios significativos. Assim, o algoritmo LARS pode ser utilizado para selecionar um subconjunto ótimo dos neurônios significativos (L_s) que minimizam (2.14) para um dado valor de λ .

Baseado nesses conceitos Bach (2008) propôs o algoritmo de seleção de variáveis *Bolasso* (*bootstrap-enhanced least absolute shrinkage operator*) que executa o algoritmo

LARS em várias replicações de *bootstrap* do conjunto de dados de treinamento. Para cada repetição, um subconjunto distinto dos regressores é selecionado.

Os passos do algoritmo *Bolasso* que são utilizados por um framework desenvolvido para o Matlab estão listados no Quadro 3 a seguir:

QUADRO 3 – Algoritmo Bolasso.

Algoritmo 1- Bolasso (*bootstrap-enhanced least absolute shrinkage operator*)

Seja n o número de exemplos, (linhas) em \mathbf{X} :

- (1) Amostrar n exemplos de (\mathbf{X}, \mathbf{Y}) , de forma uniforme e com substituição.
- (2) Calcula-se as estimativas de lasso dos pesos \mathbf{w} ao longo do caminho completo para regularização (Usando *Lars*).
- (3) Determinar quais os pesos são diferentes de zero dado um valor de λ .
- (4) Repetir os passos de 1: 3, para um número especificado de *bootstraps*.
- (5) Pegar a intersecção dos índices dos pesos diferentes de zero de todas as replicações de *bootstrap*. Selecionar as variáveis resultantes.
- (6) Readaptar usando as variáveis selecionadas via regressão de mínimos quadrados não regularizados, (se for solicitado).
- (7) Repetir o procedimento para cada valor de bt bootstraps e λ
- (8) Determinar valores "ótimos" para λ e $nbootstraps$.

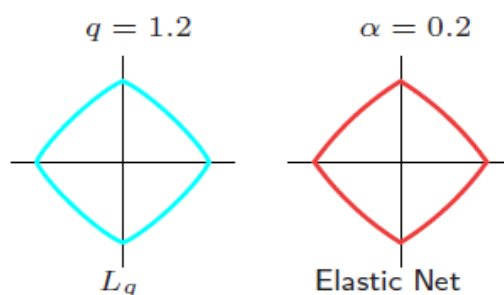
2.10 Elastic net

Quando analisamos a equação (2.13) é possível que inúmeros valores de q sejam utilizados para construir regiões de restrição. Infinitos valores diferentes de 0, 1 ou 2 podem ser utilizados, porém de acordo com Hastie (2009) o esforço para estimar o valor de q , principalmente baseado nos dados do problema, na prática, é inviável devido a variância presente no processo. Valores de $q \in [1, 2]$ sugerem um compromisso entre as regressões *ridge* e *lasso*. Em parte devido a esse fato, bem como para o trato computacional, foi

proposto por Zou e Hastie (2005) o método de penalidade *elastic net*. Esse método pode ser definido como:

$$\lambda \sum_{j=1}^p (\alpha \beta_j^2 + (1 - \alpha) |\beta_j|) \quad (2.15)$$

um compromisso diferente entre *ridge* e *lasso*. Podemos comparar $\sum_j |\beta_j|^q$ com valores obtidos através da equação (2.13). A Figura 18 mostra que quando o valor de $q = 1.2$ e o valor de $\alpha = 0.2$ na equação (2.13) os contornos são muito parecidos, onde existe uma dificuldade visual de se encontrar diferenças. O *elastic net* seleciona variáveis de forma semelhante ao método *lasso* e junto encolhe os coeficientes dos preditores correlacionados da mesma forma como é efetuado na regressão *ridge*, além de vantagens computacionais quando comparado a esses métodos de regularização (Hastie et al, 2009).



Fonte: Hastie et al (2009).

Figura 18- Contornos dos valores da constante $\sum_j |\beta_j|^q$ para $q = 1.2$ e $\alpha = 0.2$ para valores em $\lambda \sum_{j=1}^p (\alpha \beta_j^2 + (1 - \alpha) |\beta_j|)$.

2.11 Regularização MultiObjetivo

Algoritmos de aprendizado supervisionado em geral visam minimizar a soma do erro quadrático dos dados de treinamento, porém somente esse critério pode produzir uma generalização ruim em modelos que possuem dados ruidosos. Em geral, algoritmos que realizam poda em modelos de redes neurais artificiais reduzem a norma dos vetores do peso da rede. Buscando unir essas duas características de Albuquerque Teixeira et al (2000) propuseram um modelo que utiliza uma abordagem multiobjetivo buscando o equilíbrio entre as duas funções de custo (i) erro quadrático e (ii) norma do vetor de pesos. O algoritmo proposto busca uma boa

solução generalizada dentro do plano formado pelas funções de custo (i) e (ii) (de Albuquerque Teixeira et al, 2000).

Após a otimização multiobjetivo obter o conjunto Pareto Ótimo (Vira & Haimes, 1983) que contém o conjunto de soluções eficientes \check{X}^* , o próximo passo é selecionar a solução mais adequada dentro do Conjunto Pareto Ótimo. Nesse modelo é utilizado uma variação do problema e-restrito adaptado, proposto por Takahashi et al (1997). O algoritmo tem seu funcionamento baseado na impedimento de geração de soluções não viáveis, aumentando sua eficiência. Sua solução é apresentada a seguir (de Albuquerque Teixeira et al, 2000):

$f^* \in \mathbb{R}^s$ é o vetor objetivo correspondente a “solução utópica” (Vira & Haimes, 1983) do problema.

$f_i^* \in \mathbb{R}^s$, $i=1, \dots, s$, é o vetor formado pela solução ótima do objetivo individual i e os valores correspondentes a outras funções objetivo.

β é a seção de soluções restritas (cone) geradas pelos vetores $(f_i^* - f^*)$ com origem em f^* .

$\zeta_k \in \beta$ é um vetor construído de acordo com a equação 2.16 que executa uma combinação convexa dos vetores de objetivos individuais (de Albuquerque Teixeira et al, 2000).

$$\zeta_k = f^* + \check{Y}_k (f_1^* - f^*) + (1 - \check{Y}_k)(f_2^* - f^*) \text{ para } \check{Y}_k > 0 \quad (2.16)$$

A equação 2.16 sempre resulta dentro de um cone de soluções viáveis. O problema multiobjetivo pode ser visto como um problema mono-objetivo, considerando os objetivos múltiplos como restrições sobre o algoritmo de otimização. Esse problema foi resolvido utilizando um método de otimização restrito chamado de “*ellipsoid algorithm*” (Shor, 1977). O problema multiobjetivo pode ser escrito como (de Albuquerque Teixeira et al, 2000):

$$w^* = \underset{w, \eta}{arg \min} \eta \quad (2.17)$$

$$\text{s.t. } f_i(w) \leq f^* + \eta \zeta_k \quad (2.18)$$

$$f_1(\mathbf{w}) = \frac{1}{2} \sum_{j=1}^n (d_j - y_j)^2 \quad (2.19)$$

$$f_2(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2, \quad (2.20)$$

$$y_j = f_a(\mathbf{w}, x) \quad (2.21)$$

onde \mathbf{w} é o vetor de pesos de uma rede neural artificial, η é a variável auxiliar, n é o tamanho da base de treinamento d_j e y_j são respectivamente a saída desejada e saída corrente da iteração j e f_a é a função de ativação do nó. Ao substituir as equações 2.21 e 2.20 em 2.19 temos o problema restrito por equações (de Albuquerque Teixeira et al, 2000):

$$\mathbf{w}^* = \underset{\mathbf{w}}{\operatorname{arg\,min}} \underset{\mathbf{w}, \eta}{\eta} \quad (2.22)$$

$$\text{s. t.} \begin{cases} g_1(\mathbf{w}, \eta) = \frac{1}{2} \sum_{j=1}^n (d_j - y_j)^2 - f_1^* - \eta \zeta_{k1} \geq \mathbf{0}, \\ g_2(\mathbf{w}, \eta) = \frac{1}{2} \|\mathbf{w}\|^2 - f_2^* - \eta \zeta_{k2} \leq \mathbf{0} \end{cases} \quad (2.23)$$

2.12 Modelos Regularizados

Na literatura inúmeras técnicas de regularização já foram utilizadas para realizar generalizações em modelos de redes neurais, onde podemos destacar modelos que utilizam a máquina de aprendizado extremo (ELM) para realizar o treinamento da rede. As técnicas de regularização aplicadas às redes neurais têm o objetivo de aumentar a generalização e ou definir a estrutura do modelo.

Estudos foram realizados para utilizar o método *lasso*, *ridge* e *elastic net* para regularizar a ELM quando elas atuam em problemas de regressão, classificação ou reconhecimento de padrões. A principal ideia proposta por Martínez-Martínez et al (2011) é identificar o grau de relevância do peso que liga o k -ésimo elemento oculto com a camada de saída da ELM através de métodos de regressão, em especial os métodos regularizados de regressão por mínimos quadrados, que realizam penalizações ao vetor de coeficientes. A partir da aplicação dos métodos regularizadores é possível identificar e remover da

camada oculta da máquina de aprendizado extremo informações irrelevantes ou de baixa relevância para o modelo, aumentando sua capacidade de generalização. Esse método é aplicado para modelos de ELM que contém um número de pesos na camada escondida maior ou igual à quantidade de amostras de treino. Após a obtenção aleatória dos pesos da camada oculta são encontradas as respostas dos seus neurônios e usando uma abordagem regularizada são evidenciados os pesos mais relevantes que conectam a camada oculta e a camada de saída (Martínez-Martínez et al, 2011).

Um método elaborado para regularizar a ELM baseada nas regressões *ridge*, *lasso* e *elastic net* foi desenvolvido por Escandell-Montero et al (2012), porém esse processo é baseado na formação de comitês para atuar na máquina de aprendizado extremo. Essa metodologia vem atuar no problema de que alguns parâmetros permanecem inalterados após seus valores serem escolhidos aleatoriamente, pois em muitos casos eles não traduzem um valor ótimo para as operações que serão realizadas pela ELM. A proposta para resolver esse problema resume-se em utilizar um conjunto de redes ELM nos quais seus parâmetros são inicializados de forma autônoma e a combinação de suas previsões é responsável pela saída final do modelo. Esse trabalho faz uso de métodos de regularização, a fim de selecionar automaticamente os membros que farão parte do comitê (Escandell-Montero et al, 2012).

Um comitê consiste em um método que toma uma combinação de vários modelos para formar um único modelo novo. Quando tratamos de uma combinação linear, o comitê de aprendizagem do algoritmo tenta formar um conjunto de modelos e escolhe coeficientes para combiná-los. A ideia principal do método proposto reside no cálculo dos coeficientes que combinam os membros do comitê usando uma versão regularizada de regressão por mínimos quadrados. A regressão regularizada é útil, neste contexto, devido à sua tendência a preferir soluções com menores valores de parâmetros diferentes de zero, reduzindo efetivamente o número de membros do comitê (Escandell-Montero et al, 2012).

Outros métodos de regularização também podem atuar em modelos de redes neurais buscando a escolha dos melhores parâmetros para compor a arquitetura da rede. A regularização baseada em métodos estatísticos para identificar o grau de relevância dos nós da camada oculta da ELM evita que poucos nós, ou em excesso, tragam problemas para operações de classificação de padrões. Rong et al (2008) propuseram um algoritmo que identifica o grau de relevância entre os nós da camada oculta da máquina de aprendizado extremo e os rótulos de classes usando algumas técnicas estatísticas que possibilitam a

remoção das informações desnecessárias (menor relevância). Para realizar essa seleção dos atributos mais importantes da ELM é realizada a geração dos pesos da camada interna de forma aleatória. São obtidas também as respostas de cada um dos nós e juntamente com os rótulos dos dados de treinamento é calculada a relevância estatística de cada nó da camada escondida para a contribuição de se encontrar o verdadeiro rótulo da classe (Rong et al, 2008). Os métodos estatísticos utilizados nesse trabalho foram o *Chi-Squared* (Liu et al, 1995) e o *Information Gain* (Ash, 1965).

Já Miche et al (2010) modificaram o algoritmo original da ELM para criar um modelo (OP-ELM) que realiza uma poda de forma otimizada dos neurônios na camada oculta de uma máquina de aprendizado extremo. No geral o modelo é realizado em três etapas: O primeiro passo da metodologia OP-ELM é a construção real da rede neural *feedforward* (*Multilayer Perceptron*) de camada simples escondida utilizando o algoritmo de ELM original com uma série de neurônios. No segundo passo é utilizado o algoritmo *Multiresponse Sparse Regression* (Similä & Tikka, 2005) que é capaz de selecionar os neurônios mais significativos ao modelo e por fim, um processo *Leave One Out* escolhe a quantidade ideal de neurônios na camada oculta (Miche et al, 2010).

O trabalho de Huang et al (2014) usa a ELM regularizada para problemas de classificação de padrões e regressão utilizando o aprendizado semi supervisionado e não supervisionado. De uma forma diferente aos demais trabalhos correlatos eles introduzem uma forma de regularização chamada de *manifold regularization* (Belkin et al, 2006) possibilitando a sua utilização em classificação multiclasse ou em agrupamento multicluster. A proposta do ELM semi supervisionado incorpora a regularização múltipla para aproveitar os dados não rotulados para melhorar a precisão da classificação quando eles são escassos. No cenário de treinamento sem supervisão o alvo é encontrar a estrutura de dados adjacentes aos dados originais (Huang et al, 2014).

Quando a máquina aprendizado extremo atua no reconhecimento de padrões ele também pode ser regularizado. Peng et al (2013) elaboraram um método que se baseia na ideia de que amostras similares em uma ELM devem compartilhar propriedades semelhantes, formando um gráfico discriminativo regularizado para máquinas de aprendizagem extrema. Nesse método, para regularizar modelos baseados em ELM para reconhecimento facial, a restrição imposta aos pesos de saída obriga a saída das amostras a partir da mesma classe a serem semelhantes. Essa restrição é formulada com um termo de regularização que é adicionado no objetivo básico de um modelo de ELM, fazendo com

que os pesos da camada de saída também sejam resolvidos de forma analítica (Peng et al, 2013).

Por fim o modelo de Zhou et al (2014) para identificar o sistema de variação de tempo é baseado em máquina de aprendizado extremo regularizada com um mecanismo de esquecimento chamado FORELM além de um dispositivo ELM *kernel-online*. Nesse trabalho o modelo idealizado foi construído através dos conceitos de FOS-ELM (Zhao et al, 2012) e ReOS-ELM (Huynh & Won, 2011) . O algoritmo FORELM é capaz de superar o problema potencial das matrizes de singularidade utilizando a regularização para eliminar os efeitos dos dados desatualizados do modelo incorporando mecanismos de esquecimento.

Quando as redes neurais *feedforward* de camada simples escondida (SLFN) são utilizadas para modelar sistemas de variável no tempo, amostras de treinamento não só são geradas uma a uma, mas muitas vezes também têm propriedades atuais; ou seja, os dados de formação têm um prazo de validade. Por isso, durante o processo de aprendizagem on-line pelo algoritmo sequencial de aprendizagem, os dados de treinamento mais antigos ou desatualizados, cuja eficácia é menor ou é perdida após várias unidades de tempo, deve ser abandonada, que é a ideia de mecanismo de esquecimento. ReOS-ELM e KB-IELM (Guo, Hao & Liu, 2014) podem não refletir a atualidade dos dados de formação sequenciais. O mecanismo de esquecimento é adicionado a eles para eliminar os dados desatualizados que podem ter efeito enganoso ou ruim no SLFN construído.

Técnicas de regularização também foram aplicadas em modelos adaptativos. Nesse caso foram aplicadas técnicas multiobjetivo que buscavam além da regularização do modelo a diminuição do erro de treinamento. Carrano et al (2008) utilizaram técnicas de algoritmos genéticos buscando melhorar a resposta do modelo e ao mesmo tempo diminuir a complexidade da rede resultante de um modelo ANFIS. Buscando resolver o dilema do “*bias-variance*” foi proposto um modelo de treinamento multiobjetivo que considera a complexidade da rede conjuntamente com o erro sobre o conjunto de treinamento como objetivos relevantes buscando a minimização de ambos. A abordagem multiobjetivo proposta cria e evolui todo o conjunto de modelos simultaneamente fazendo o possível para avaliar os modelos com complexidades distintas dentro do processo de formação, sem a necessidade de um *loop* extra para adaptar a estrutura da rede. Os algoritmos utilizados para realizar o treinamento do ANFIS foram o NSGA- II (Deb et al , 2002) com o *real-based encoding and operators* (Ramos et al, 2003) (Carrano et al, 2008).

3 Rede Neural Nebulosa para classificação binária

Neste capítulo serão apresentados os métodos e procedimentos adotados para o desenvolvimento da rede neural nebulosa proposta.

Esse trabalho propõe uma nova metodologia de aprendizado para redes neurais nebulosas do tipo *feedforward* com uma camada de fuzzificação, uma camada escondida, composta por neurônios nebulosos, e uma camada de saída linear. O algoritmo proposto é capaz de gerar modelos consistentes e precisos agregando interpretação à estrutura resultante.

O aprendizado do modelo é realizado através de conceitos de máquina de aprendizado extremo, porém um termo regularizador é adicionado na função de custo que, juntamente com uma técnica de reamostragem, é capaz de realizar a seleção dos melhores neurônios em camadas internas, gerando modelos parcimoniosos.

Inicialmente, conjuntos nebulosos, com funções de pertinência igualmente espaçadas, são definidos para cada variável de entrada na camada de fuzzificação. Posteriormente é gerado um conjunto inicial de neurônios nebulosos candidatos. A partir desse conjunto inicial de neurônios, o algoritmo *bootstrap lasso* (Bach, 2008) é utilizado para definir a topologia da rede, selecionando um subconjunto dos neurônios nebulosos significativos. Finalmente o algoritmo de mínimos quadrados é utilizado para estimar os pesos da camada de saída da rede.

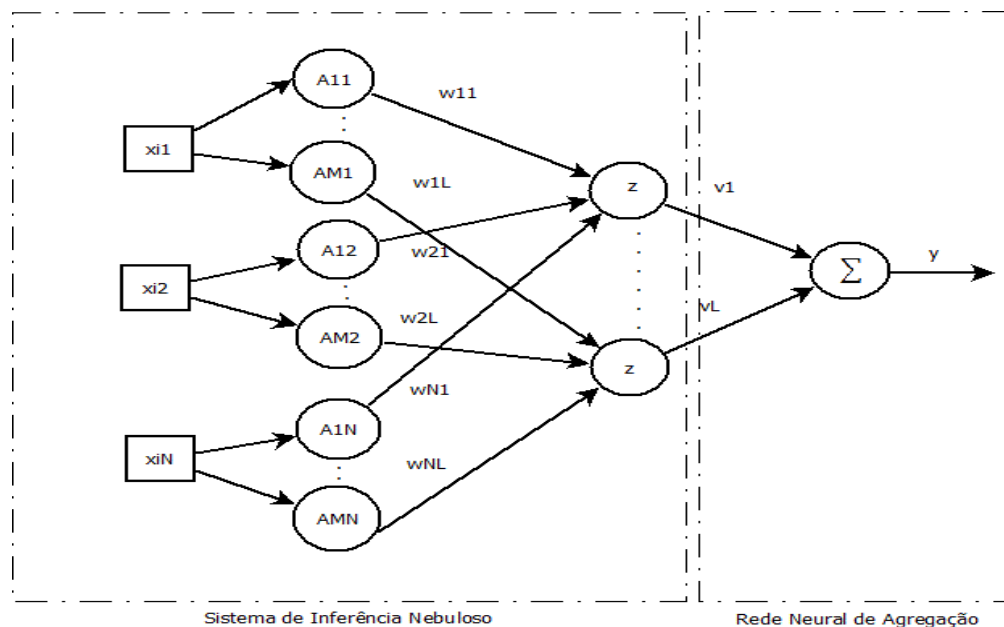
Conforme mencionado anteriormente vamos detalhar a estrutura da rede neural nebulosa discriminando suas camadas e metodologias empregadas para construção da rede.

3.1 Neurônios Lógicos Nebulosos Utilizados

As redes neurais nebulosas discriminadas a seguir utilizam os dois tipos dos neurônios lógicos nebulosos descritos anteriormente. Os neurônios lógicos que compõem a rede são descritos em (2.7) e (2.9), chamados aqui de *andneuron* e *unineuron* respectivamente. A uninorma utilizada para a construção do unineuron está representada em (2.5).

3.2 Rede Neural Nebulosa.

A estrutura da rede proposta é ilustrada na Figura 19 em que os neurônios z podem ser *andneurons* ou *unineurons*.



Fonte: Dados da própria Pesquisa

Figura 19- Rede Neural Nebulosa Proposta.

A primeira camada é definida como camada de fuzzificação e é composta por neurônios cujas funções de ativação são funções de pertinência dos respectivos conjuntos nebulosos utilizados na partição das variáveis de entrada. Para cada variável de entrada x_j são definidos M conjuntos nebulosos A_{mj} , para m variando de $1, \dots, M$. As saídas da primeira camada são os graus de pertinência associados com os valores de entrada, isto é, $a_{jm} = \mu_{A_{mj}}^{x_j}$ para $j=1, \dots, N$ e $m=1, \dots, M$, onde N é o número de entradas e M é o número de conjuntos nebulosos para cada variável de entrada.

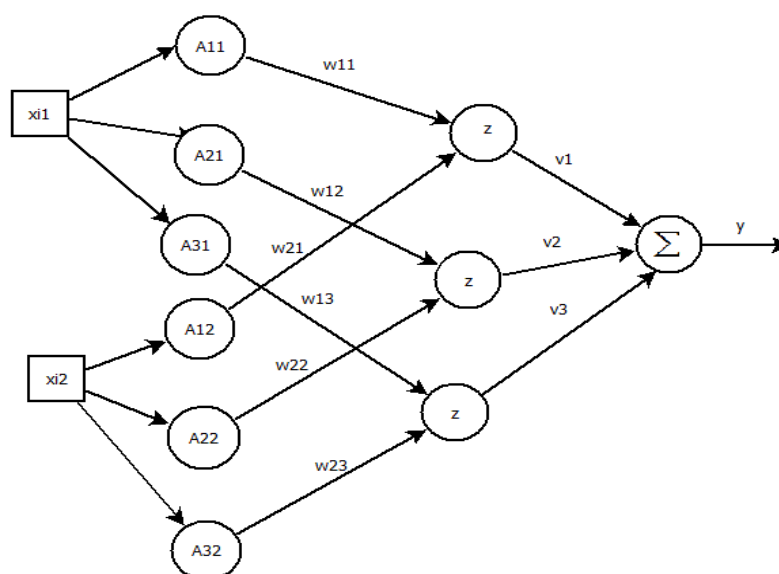
A segunda camada é composta de L neurônios lógicos nebulosos, do tipo *andneuron* ou *unineuron*. Cada neurônio executa uma agregação ponderada de algumas saídas da primeira camada. Os neurônios lógicos nebulosos realizam a agregação utilizando os pesos w_{il} (para $i=1, \dots, N$, e $l=1, \dots, L$). Para cada variável de entrada j , apenas uma saída a_{jm} é definida como entrada do l -ésimo neurônio. Ademais, buscando gerar topologias de redes difusas, cada neurônio da segunda camada está associado a uma variável de entrada, ou seja, a matriz de pesos w é esparsa.

Por fim, a camada de saída é composta por um único neurônio linear. Para problemas de classificação de padrões binários, em que as saídas são definidas como $\{-1, 1\}$. A saída do modelo é dada por:

$$y = \text{sign}(\sum_{j=0}^l z_j v_j) \quad (3.6)$$

onde $z_0 = 1$, v_0 é o bias, e z_j e v_j , $j = 1, \dots, l$ são a saída de cada neurônio nebuloso da segunda camada e o seu peso correspondente, respectivamente.

Regras nebulosas podem ser extraídas da topologia da rede. A Figura 20 ilustra o processo de extração de regras a partir da estrutura da rede. A rede desse exemplo possui duas variáveis de entrada, três conjuntos nebulosos para cada variável e três neurônios lógicos nebulosos do tipo *andneuron*, ou seja: $N=2$, $M=3$ e $L=3$. As seguintes regras se/então podem ser extraídas da rede:



Fonte: Dados da própria Pesquisa

Figura 20- Exemplo Rede Neural Nebulosa com $N=2$, $M=3$ e $L=3$.

Regra1: Se x_{i1} é A^1_1 com certeza w_{11} ...

e x_{i2} é A^1_2 com certeza w_{21} ...

então y_1 é v_1

Regra2: Se x_{i1} é A^2_1 com certeza w_{12} ...

(3.7)

e x_{i2} é A^2_2 com certeza w_{22} ...

então y_2 é v_2

Regra3: Se x_{i1} é A^3_1 com certeza w_{13} ...

e x_{i2} é A^3_2 com certeza w_{23} ...

então y_3 é v_3

A seguir podemos verificar de forma mais detalhada como todos os passos do treinamento do modelo acontecem de forma mais detalhada.

3.3 Treinamento da rede neural nebulosa utilizando Máquina de Aprendizado Extremo.

Lemos et al (2010) propôs uma metodologia de treinamento para redes neurais nebulosas em que um algoritmo de agrupamento é utilizado para definir a topologia da rede (neurônios das camadas de fuzzificação e escondida, e as respectivas conexões) e um algoritmo genético é utilizado para estimar seus parâmetros. Apesar de o modelo apresentar uma boa acurácia nos problemas abordados, este possuía um elevado custo computacional.

Buscando resolver o problema, em Lemos et al (2012) a etapa de ajustes dos parâmetros é realizada por uma metodologia mais rápida e eficiente baseada em conceitos de máquina de aprendizado extremo. A ELM é um algoritmo de aprendizagem desenvolvido para redes neurais *feedforward* de camada escondida (SLNFs), onde são atribuídos valores aleatórios para os pesos da primeira camada e os pesos da camada de saída são estimados analiticamente. Na nova versão de treinamento para a rede neural nebulosa, os parâmetros dos neurônios são atribuídos de forma aleatória e os parâmetros de saída são calculados através de mínimos quadrados.

Esse modelo reduz o custo computacional do treinamento, porém, assim como a primeira metodologia, a topologia da rede é definida utilizando um algoritmo de agrupamento. Essa abordagem possui uma desvantagem de gerar modelos que não são facilmente interpretáveis, ou seja, apesar de regras poderem ser extraídas da topologia final da rede, os conjuntos nebulosos estimados para cada variável de entrada, muitas vezes, não possuem um significado linguístico.

Esse trabalho propõe uma nova abordagem para o treinamento de redes neurais nebulosas que aborda o problema de interpretabilidade dos resultados. Propõe-se a

definição de funções de pertinência igualmente espaçadas para cada variável de entrada, para definir os neurônios da camada de fuzzificação e a utilização de uma técnica de regularização para definir a topologia da camada escondida. O algoritmo proposto é capaz de gerar modelos parcimoniosos, selecionando neurônios mais relevantes dentro do contexto do problema. A partir do modelo resultante é possível extrair um conjunto de regras nebulosas, conforme visto em (3.7).

O algoritmo de aprendizagem proposto inicialmente define os neurônios da primeira camada através da partição de cada intervalo de cada variável de entrada em M conjuntos nebulosos com funções de pertinência gaussianas igualmente espaçadas. Em seguida, uma estratégia de partição do espaço de entrada por um *grid* (Jang, 1993) é utilizada para definir um conjunto inicial de neurônios candidatos. O número inicial de neurônios na camada escondida é definido como M^N , ou seja, para cada possível combinação das funções de pertinência de cada entrada, um neurônio é gerado e suas entradas são definidas. Os pesos associados às entradas do neurônio são definidos aleatoriamente no intervalo $[0,1]$, de forma similar às ELMs.

Por fim a arquitetura final da rede é definida através de uma técnica de extração de características baseada em regularização L_1 e reamostragem. O algoritmo de aprendizagem assume que a saída da camada escondida composta pelos neurônios candidatos pode ser escrita como:

$$f(x_i) = \sum_{l=0}^{L_\rho} v_l z_l(x_i) = \mathbf{z}(x_i) \mathbf{v} \quad (3.8)$$

onde $\mathbf{v}=[v_0, v_1, v_2, \dots, v_{L_\rho}]$ é o vetor de pesos da camada de saída e o $\mathbf{z}(x_i) = [z_0, z_1(x_i), z_2(x_i), \dots, z_{L_\rho}(x_i)]$ é o vetor de saída da segunda camada, para $z_0=1$. Nesse contexto, $\mathbf{z}(x_i)$ é considerado como o mapeamento não linear do espaço de entrada para um espaço de características nebulosas de dimensão L_ρ . Como os pesos que ligam as duas primeiras camadas são atribuídos de forma aleatória, os únicos parâmetros a serem estimados são os pesos da camada de saída. Dessa forma, o problema de estimativa dos parâmetros da rede pode ser visto como um problema de regressão linear simples, permitindo a utilização de técnicas de regressão (Hastie et al, 2009) para estimativa dos parâmetros e seleção dos neurônios candidatos.

3.3.1 Geração de funções de pertinência

A granulação do espaço de entrada é concretizada através de uma partição em grid dos dados (sem agrupamento) (Jang, 1997). As funções de pertinência adotadas nesse trabalho (M) são assumidas como gaussianas e elas são geradas de forma igualmente espaçada no espaço de entrada. O número de neurônios da primeira camada (L) é igual à quantidade de funções de pertinência elevado pelo número total de variáveis de entrada, ou seja,

$$L = M^N \quad (3.9)$$

Para um problema que possui quatro variáveis de entrada e três funções de pertinência para cada uma das variáveis geramos um total de $3^4=81$ neurônios nebulosos. Na Figura 21 a seguir podemos visualizar as funções gaussianas igualmente espaçadas.

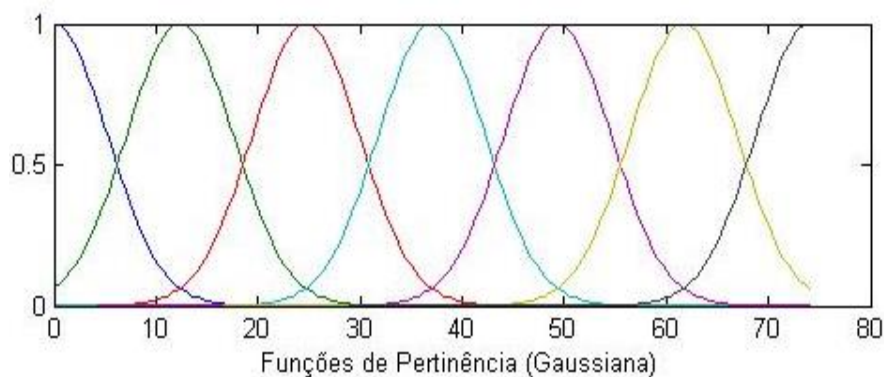


Figura 21- Funções de pertinência gaussianas igualmente espaçadas.

Fonte: Dados da própria Pesquisa

Essa abordagem de definição da topologia da rede facilita a interpretabilidade das regras extraídas. Por exemplo, caso sejam utilizados três conjuntos nebulosos por variável de entrada (M=3) cada conjunto nebuloso pode ser interpretado pelos valores linguísticos: “Pequeno”, “Médio” e “Grande”.

Essa metodologia apresenta um problema de crescimento exponencial do número de neurônios gerados em função do número de variáveis de entrada (Jang, 1993). Por exemplo, caso M=3 e N=7, inicialmente 2187 neurônios serão definidos. Para contornar esse problema, uma estratégia híbrida é utilizada nesse trabalho. Caso o número de variáveis de entrada seja menor ou igual a seis ou a quantidade de neurônios gerados por

qualquer uma das combinações entre M e N seja maior do que 729 neurônios utiliza-se a estratégia baseada em *grid*². Caso contrário utiliza-se uma estratégia que gera aleatoriamente um subconjunto de tamanho limitado (nesse trabalho o tamanho foi definido como o dobro de amostras com o limite máximo de 729) das possíveis combinações presentes no *grid*.

Em seguida, um filtro é aplicado para remoção de neurônios que estejam associados a regiões do espaço de entrada que possuam poucas ou nenhuma amostra. Para isso calcula-se a saída de cada um dos L neurônios para todas as amostras do conjunto de treinamento. Para cada neurônio, somam-se os valores das suas respectivas saídas. Finalmente, os neurônios são ordenados decrescentemente de acordo com o valor da soma e os primeiros 200 neurônios são selecionados formando assim um subconjunto de neurônios candidatos, chamado L_p . Nesse trabalho, essa técnica de filtragem só é aplicada caso o número inicial de neurônios gerados pela estratégia de particionamento do espaço de entrada por um *grid* seja maior ou igual a 200. Caso contrário, o número de neurônios candidatos (L_p) será o mesmo número de neurônios gerados pela partição em *grid*.

3.3.2 Procedimento de regularização da rede

Uma abordagem existente para aumentar a estabilidade do algoritmo de seleção de modelo é a utilização de reamostragem. Nessa dissertação o *bootstrap lasso* é utilizado para definir a topologia final da rede, escolhendo os neurônios lógicos nebulosos mais significativos, chamados de L_s .

Após a escolha dos neurônios candidatos (L_p) utiliza-se o algoritmo Bolasso, relatado na Seção 2.9.1, que é executado nos neurônios significativos da rede neural nebulosa buscando através de replicações realizadas com várias combinações de L_p encontrar os neurônios que tem maior relevância para o problema atual. Essas replicações geram uma lista de regressores em cada uma das replicações, onde são escolhidos os melhores neurônios lógicos nebulosos conforme os parâmetros utilizados pelo algoritmo. A quantidade de replicações e o valor de consenso são exemplos de valores que podem interferir na escolha dos neurônios mais importantes. Nesse trabalho é utilizada a validação cruzada (Kohavi et al, 1995) nos dados de treinamento para encontrar o melhor conjunto de variáveis selecionadas.

² Disponível em J.R. Jang, Frequently asked questions – ANFIS in the fuzzy logic toolbox, <<http://www.cs.nthu.edu.tw/~jang/anfisfaq.htm>> Acesso em Agosto/2015.

Os regressores a serem incluídos no modelo final são definidos de acordo com a frequência com que cada um deles é selecionado através de diferentes ensaios (Bach, 2008). Um limiar de consenso é definido, digamos $\gamma = 50\%$, e um regressor está incluído, se for selecionado como importante em, pelo menos, 50% dos ensaios.

3.3.3 Atualização de pesos da rede neural artificial de agregação e utilização da Máquina de Aprendizado Extremo

Por fim, após a definição da topologia da rede é realizado o cálculos da estimativa do vetor de pesos da camada de saída. Nesse trabalho esse vetor é estimado pela pseudo Inversa de Moore-Penrose (Penrose, 1955) utilizando as saídas de todos os neurônios lógicos nebulosos mais significativos da segunda camada, \mathbf{z} :

$$\mathbf{v} = \mathbf{z}^+ \mathbf{y} \quad (3.10)$$

onde \mathbf{Z}^+ é a pseudo Inversa de Moore-Penrose de \mathbf{z} que é a norma mínima da solução dos mínimos quadrados para os pesos da saída. Podemos expressar \mathbf{z} como sendo:

$$\mathbf{z} = \begin{bmatrix} z_0 & z_1(x_1) & \cdots & z_{L_S}(x_1) \\ z_0 & z_1(x_2) & \cdots & z_{L_S}(x_2) \\ \vdots & \vdots & \cdots & \vdots \\ z_0 & z_1(x_n) & \cdots & z_{L_S}(x_n) \end{bmatrix} \quad (3.11)$$

O processo de aprendizagem pode ser sintetizado na escolha desses três parâmetros:

- o número de conjuntos nebulosos que vão particionar o espaço de entrada, M ;
- o número de replicações *bootstrap*, bt ;
- o limiar de consenso, γ .

O quadro 4 a seguir mostra os principais passos a serem executados pelo modelo de treinamento proposto:

QUADRO 4 – Algoritmo de Aprendizado Proposto para Redes Neurais Nebulosas

Algoritmo 2- Algoritmo de Aprendizado Proposto para Redes Neurais Nebulosas
<p>Define M conjuntos nebulosos para cada variável de entrada</p> <p>Para o conjunto de treinamento (X e Y)</p> <p>Se $N \leq 6$ ou $M^N \leq 729$ então</p> <p>Utiliza a partição em grid com M funções de pertinência igualmente espaçadas para cada variável de entrada extraíndo todas as combinações de regras formadas (L).</p> <p>Senão</p> <p>Utilizar a partição em grid com funções de pertinência igualmente espaçadas gerando combinações de forma aleatória para gerar um subconjunto limitado de regras com limite de 729 combinações (L).</p> <p>Fim senão</p> <p>Se $L > 200$ então</p> <p>Aplica filtro de seleção de 200 regras mais relevantes (L_ρ).</p> <p>Senão</p> <p>Neurônios candidatos (L_ρ) = L.</p> <p>Fim Senão</p> <p>Para todas N entradas faça</p> <p>Calcular o mapeamento $\mathbf{z}(x_i)$ utilizando AndNeuron / Unineuron</p> <p>fim para</p> <p>Define a quantidade de replicações (bt) e o valor de consenso (γ) <i>para o algoritmo bootstrap lasso</i></p> <p>Selecione L_s neurônios significativos utilizando <i>bootstrap lasso</i>.</p> <p>Estima os pesos da camada de saída (3.10).</p>

4 Experimentos Numéricos

Nesse capítulo será apresentada uma análise dos dados coletados nos testes de classificações de bases binárias. Para demonstrar como foram realizados os experimentos vamos discriminar os fatores de teste, a forma de avaliação, os algoritmos e bases utilizadas além dos resultados encontrados. Por fim uma análise estatística auxiliará na validação e comparação do modelo com demais métodos na literatura.

4.1 Características do Problema e Forma de Avaliação

O modelo de aprendizagem de redes neurais nebulosas propostos nesse trabalho foi avaliado através de testes de classificação de padrões binários.

A acurácia do modelo ao classificar as bases binárias é calculada através da comparação do valor obtido pelo modelo, confrontando-o com o valor esperado. A cada acerto do neurônio (valor esperado é igual ao valor calculado) é incrementado uma unidade. No final é dividida a quantidade de acertos com o total possível de respostas corretas para a situação avaliada. O valor percentual é calculado para a acurácia de cada uma das classes. As variáveis envolvidas no processo foram normalizadas com média zero e variância um. Todas as saídas do modelo foram normalizadas para o intervalo $[-1,1]$ e as funções de ativação do tipo gaussianas.

4.2 Características gerais dos algoritmos e testes executados.

O algoritmo proposto nesse trabalho foi comparado com outros classificadores para verificar sua capacidade de acurácia. Nas Tabelas 2 e 5 tabelas apresentadas a seguir, o valor principal é a acurácia média de 30 experimentos realizados e o valor entre parênteses é o respectivo desvio padrão. Já na Tabela 3 o valor principal é a média dos neurônios mais relevantes ao contexto avaliado juntamente com o valor do desvio padrão em parênteses nos 30 ensaios. Considera-se o *R-ANDNEURON* a rede formada por neurônios lógicos compostos por neurônios *and*, *R-UNINEURON* por *unineurons* e uma versão do treinamento sem a utilização da regularização, SEM-REGULARIZAÇÃO.

Foram realizados três testes buscando verificar a eficiência e interpretabilidade do modelo. No primeiro teste o *R-ANDNEURON* E o *R-UNINEURON* foram avaliados utilizando uma base sintética. Esse teste busca ilustrar a interpretabilidade da rede resultante. Já nos dois testes seguintes foram utilizadas as bases de um repositório dados e

pesquisas na área de *machine learning* (Bache et al, (2013), Hsu et al, (2003) e Ho et al (1996)) para realizar a classificação binária. No segundo teste realizado buscou-se verificar a capacidade do modelo de proposto nesse trabalho de aperfeiçoar a estrutura da rede através do método de regularização descrito no capítulo anterior. Para efetuar essas comparações, utilizou-se o R-ANDNEURON e o R-UNINEURON, a ELM em sua versão original, além de um algoritmo regularizado de poda para ELM. Esse algoritmo, proposto por Miche et al (2010) (OP-ELM), seleciona os neurônios utilizando LARS. Propõe-se analisar o desempenho conforme classes de pequenas ($N \leq 6$) e altas ($N > 6$) dimensões. O número de neurônios da ELM e o número de neurônios iniciais da OP-ELM é o mesmo número dos algoritmos R-ANDNEURON e R-UNINEURON, ou seja, L_p , e a ideia é comparar além da acurácia de cada um dos algoritmos o número de neurônios finais da rede (L_s), após a definição da topologia.

Por fim o terceiro teste é uma proposta uma comparação entre o modelo sem a utilização do método de regularização com o R-ANDNEURON e R-UNINEURON para confirmar a eficiência da regularização no modelo de treinamento proposto apenas pela acurácia ao realizar as classificações binárias de bases com menor número de dimensões.

4.3 Testes de interpretabilidade da rede resultante

4.3.1 Base de dados sintética

Para realizar o teste a seguir foi utilizada uma base sintética de dados bidimensionais buscando mostrar a possível interpretabilidade do modelo proposto. O conjunto de dados artificial é composto de 40 amostras e foi formada através de duas gaussianas com covariância igual à matriz identidade e centradas em $[0,0]$ e $[6,6]$ respectivamente. Nesse contexto também serão utilizados 70 % das amostras para treinamento e 30% para os testes. A Figura 22 representa a base sintética utilizada:

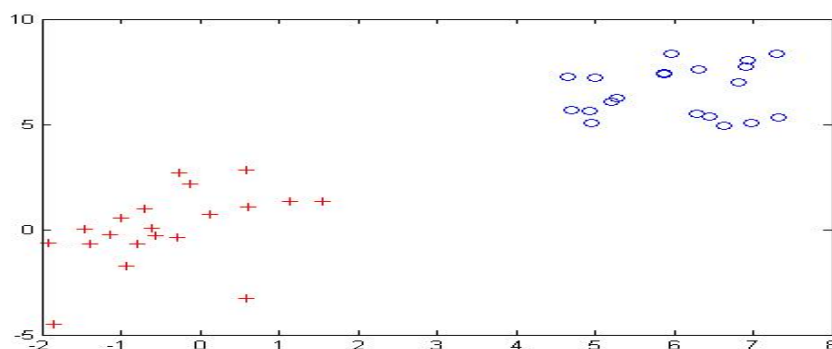


Figura 22- Base de dados sintética para problema de classificação.

A partir dessa base foram geradas as seguintes funções de pertinência ilustradas na Figura 23 para as duas variáveis do problema.

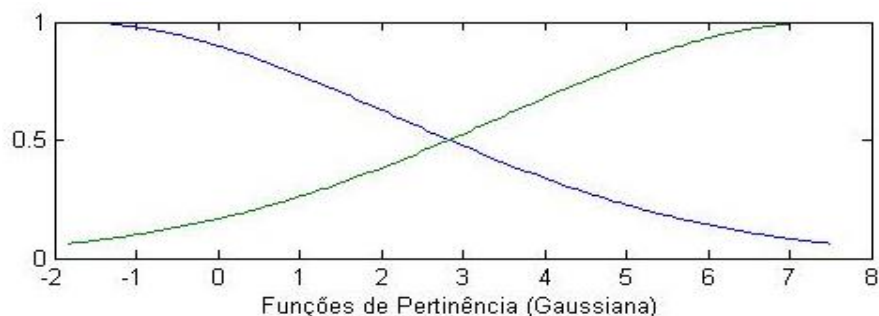


Figura 23- Funções de Pertinência geradas para a base sintética.

Fonte: Dados da própria Pesquisa

4.3.2 Resultados experimentais

No primeiro teste para o valor de $M=2$ a configuração inicial da rede neural nebulosa utilizaria quatro neurônios lógicos nebulosos. Porém, após a regularização da rede foi constatada a utilização de dois neurônios mais significativos ao problema, que são capazes de realizar corretamente a classificação de 100 % das amostras de teste. Esses dois neurônios podem ser interpretados a partir das seguintes regras:

Regra1: Se x_{i1} é Grande com certeza 0.9619

e x_{i2} é Grande com certeza 0.0046

então y_1 é 0.8854

Regra2: Se x_{i1} é Pequeno com certeza 0.7482

(4.1)

e x_{i2} é Pequeno com certeza 0.4505

então y_2 -0.3273

4.4 Testes de aperfeiçoamento da estrutura da rede.

4.4.1 Base de dados reais.

Para os dois testes a seguir foram utilizados 13 bases para avaliarmos o desempenho dos algoritmos ao classificar os dados. Para a obtenção de padrões consistentes de verificação de resultados utilizaremos bases de dados conhecidas em estudos de aprendizado de máquina disponível no UCI *machine learning repository* (Bache

& Lichman, 2013) e em algumas pesquisas de classificação de padrões realizadas por Hsu et al, (2003) e Ho et al (1996).

Ressalta-se que para fins de evitar tendências nos testes executados, todas as amostras disponíveis foram permutadas e foram coletadas 30 medições de acurácia e dos neurônios selecionados mais relevantes de cada uma das bases avaliadas em cada modelo analisado. A Tabela 1 traz as informações completas sobre as bases como o nome, as siglas utilizadas, o número de amostras e a quantidade dessas amostras destinadas para treinamento (70%) e teste (30%) em cada um dos algoritmos.

Tabela 1- Informações das Bases de Teste.

Base	ABrev.	n	Trein.	Teste
Four Class	FCL	862	603	259
Haberman	HAB	306	214	92
Iris	IRI	150	105	45
Transfusion	TRA	748	523	225
Mammographic	MAM	830	581	249
Liver Disorders	LIV	345	242	103
Pim.I. Diabetes	DIA	768	538	230
Heart	HEA	270	189	81
German Credit	GER	558	390	168
Climate	CLI	540	377	163
Aust. Credit	AUS	1000	700	300
Parkison	PAR	195	137	58
Ionosphere	ION	351	245	106

4.4.2 Aperfeiçoamento da rede resultante, resultados experimentais e análises estatísticas.

No segundo teste utilizou-se a validação cruzada (Kohavi et al, 1995) para estimar os parâmetros $M = \{3, 5\}$, $bt = \{32, 64, 128\}$ e $\gamma = \{50\%, 60\%, 70\%\}$ onde para tal são utilizadas 5 partições.

A Tabela 2 apresenta os resultados de acurácia dos métodos avaliados e na Tabela 3 os respectivos resultados de neurônios selecionados no segundo teste. Na Tabela 4 são expostos os resultados finais dos testes estatísticos referentes às Tabelas 2 e 3.

Tabela 2- Resultados de acurácia nos testes efetuados.

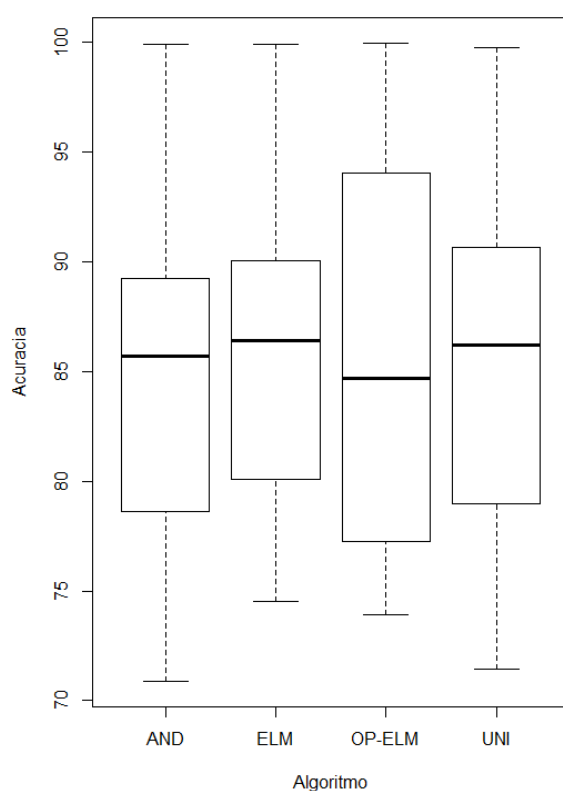
Base	R-UNINEURON	OP-ELM	R-ANDNEURON	ELM
FCL	79,95 (4,08)	76,06 (2,51)	81,21 (4,01)	75,10 (3,68)
HAB	75,12 (1,29)	75,15 (1,37)	74,17 (1,58)	74,50 (1,28)
IRI	99,75 (0,64)	99,84 (0,46)	99,91 (0,25)	99,88 (0,28)
TRA	78,03 (1,09)	79,05 (2,82)	77,06 (2,31)	76,89 (1,48)
MAM	86,22 (1,32)	84,70 (1,12)	85,67 (0,77)	82,01 (1,69)
LIV	71,45 (3,05)	73,89 (4,43)	70,90 (2,89)	80,10 (3,11)
DIA	78,97 (1,55)	77,25 (3,74)	78,62 (1,50)	81,08(4,45)
HEA	90,16 (2,44)	84,32 (3,21)	89,22 (3,26)	90,04 (2,95)
GER	90,07 (2,35)	98,34 (2,17)	89,07 (1,38)	89,22 (1,78)
CLI	91,94 (1,74)	99,94 (0,19)	94,39 (2,09)	94,08 (3,11)
AUS	80,01 (2,12)	89,20 (4,26)	78,98 (3,14)	86,42 (4,89)
PAR	96,78 (4,17)	94,02 (4,45)	97,53 (3,26)	87,40 (2,99)
ION	90,66 (4,85)	91,58 (5,47)	88,79 (7,02)	93,19 (4,21)

Tabela 3- Neurônios selecionados no teste.

Base	L_p	R-UNINEURON. (L_s)	OP-ELM. (L_s)	R-ANDNEURON (L_s)	ELM (L_s)
FCL	9(0)	7,02 (1,47)	8,10 (0,51)	7,83 (0,83)	9(0)
HAB	84,66(82,95)	11,87 (4,71)	39,31 (2,64)	9,87 (3,12)	84,66(82,95)
IRI	105,80(50,50)	31,87 (12,68)	33,52 (10,12)	28,73 (10,07)	105,80(50,50)
TRA	124,63(58,33)	21,12 (2,68)	42,68 (3,12)	18,66 (5,13)	124,63(58,33)
MAM	200(0)	26,74 (2,41)	43,12 (16,66)	22,67 (2,96)	200(0)
LIV	200(0)	34,47 (7,11)	85,54 (18,73)	14,08 (9,26)	200(0)
DIA	200(0)	27,94 (9,02)	76,36 (36,74)	24,80 (8,74)	200(0)
HEA	200(0)	53,22 (12,66)	99,77 (39,16)	46,67 (8,92)	200(0)
GER	200(0)	23,98 (12,87)	120,80 (31,98)	34,58 (18,98)	200(0)
CLI	200(0)	22,69 (14,57)	122,95 (42,12)	24,69 (10,19)	200(0)
AUS	200(0)	33,14 (13,41)	148,16 (27,05)	32,44 (6,42)	200(0)
PAR	200(0)	27,74 (4,96)	109,85 (16,98)	16,35 (7,62)	200(0)
ION	200(0)	53,27 (12,59)	107,56 (26,74)	49,87 (11,87)	200(0)

Ao analisar as Tabelas 2 e 3 constata-se que os algoritmos analisados obtiveram resultados bem próximos de acurácia na classificação das bases utilizadas no teste, porém a quantidade de neurônios selecionada (L_s) foi bem menor para o R-ANDNEURON e o R-UNINEURON em todos os casos. Para confirmar o desempenho do modelo proposto busca-

se realizar uma análise estatística sobre os dados coletados na classificação de padrões. Os algoritmos serão avaliados considerando a acurácia como único fator, onde cada um das bases de classificação pode ser vista como o fator de blocagem. Para realizar os testes estatísticos utiliza-se no teste realizado a análise de variância (ANOVA) sobre os resultados de cada um dos grupos (algoritmo x fator de bloco). De uma forma geral constata-se que o teste possui 52 (4 algoritmos e 13 bases) grupos. Através desse teste busca-se concluir se o desempenho dos algoritmos propostos nessa dissertação apresenta desempenho médio igual aos algoritmos utilizados na comparação ao realizar as classificações de padrões. A figura 24 apresenta a comparação entre as acurácias coletadas nos testes de classificação.

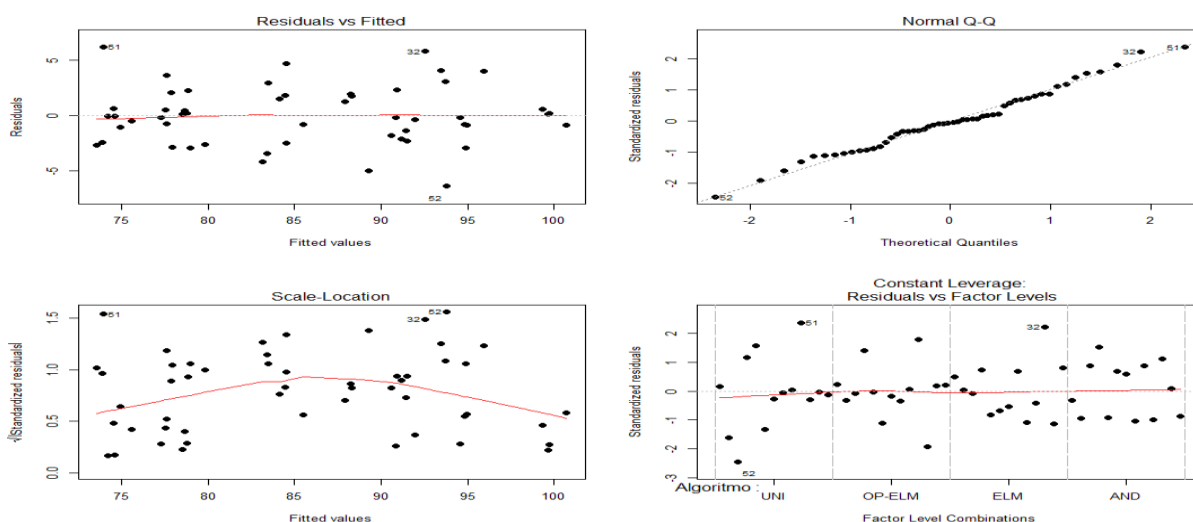


Fonte: Dados da própria Pesquisa

Figura 24- Plot do Teste 2 - Acurácia dos algoritmos analisados.

Após a execução do teste ANOVA conclui-se a um nível de significância de $\alpha=0.05$ que devemos aceitar a hipótese nula de igualdade dos desempenhos dos algoritmos a realizarem a classificação com essas bases de dados ($p\text{-value}=0.699$). Após confirmar todas as premissas de validação do resultado (normalidade ($p\text{-value}=0.8426$), homoscedasticidade ($p\text{-value}=0.5656$) e independência dos dados ($p\text{-value}=0.3665$)) que podem ser visualizadas na figura 25. Posteriormente realizou-se um teste *post-hoc* de múltiplas comparações chamado teste de *Tukey* (Hsu, 1996) que realiza comparações de

duas a duas entre todos os algoritmos envolvidos. As múltiplas comparações são apresentadas na Tabela 4 temos evidências suficientes nos resultados obtidos para aceitar a equivalência entre o *R-UNINEURON*, OP-ELM, ELM e R-ANDNEURON na classificação das 13 bases analisadas. Os resultados de múltiplas comparações ainda nos fornecem que o comportamento entre o R-UNINEURON o R-ANDNEURON e a ELM são extremamente semelhantes.



Fonte: Dados da própria Pesquisa

Figura 25- Validações do Teste ANOVA- Teste 2.

Portanto o modelo proposto pode ser visto como uma metodologia viável para realizar classificação de dados reais.

Tabela 4- Resultados das múltiplas comparações de Tukey.

```

Tukey multiple comparisons of means
 95% family-wise confidence level
Fit: aov(formula = Acuracia ~ Algoritmo + Problema, data =
data)

$Algoritmo
          diff      lwr      upr    p adj
AND-OP-ELM -1.36769231 -4.674222  1.938837 0.6833800
ELM-OP-ELM -1.03307692 -4.339606  2.273453 0.8343388
UNI-OP-ELM -1.09461538 -4.401145  2.211914 0.8092226
ELM-AND     0.33461538 -2.971914  3.641145 0.9927976
UNI-AND     0.27307692 -3.033453  3.579606 0.9960460
UNI-ELM    -0.06153846 -3.368068  3.244991 0.9999539

```

4.5 Testes de eficiência do modelo de treinamento regularizado.

4.5.1 Resultados experimentais e análises estatísticas.

O terceiro teste considera $2 \leq M \leq 5$, $bt=32$ e $\gamma=50\%$ para verificar-se se o modelo de treinamento regularizado proposto para as redes neurais nebulosas mantém a eficiência em classificar bases de dados.

A Tabela 5 apresenta os resultados de comparação entre os resultados dos modelos regularizados e o modelo proposto pelo trabalho quando não é aplicado sobre as bases o método de regularização *Bolasso*. Como o método sem regularização possui limitações quanto à quantidade de neurônios utilizadas os testes foram realizados com poucas funções de pertinência para cada uma das variáveis aplicadas ao modelo. Esse número varia entre duas até o número de cinco funções de pertinências (F. P.) conforme apresentados na tabela a seguir. Cada método apresenta uma quantidade diferente de neurônios candidatos e neurônios utilizados, portanto o foco desse teste será a acurácia encontrada.

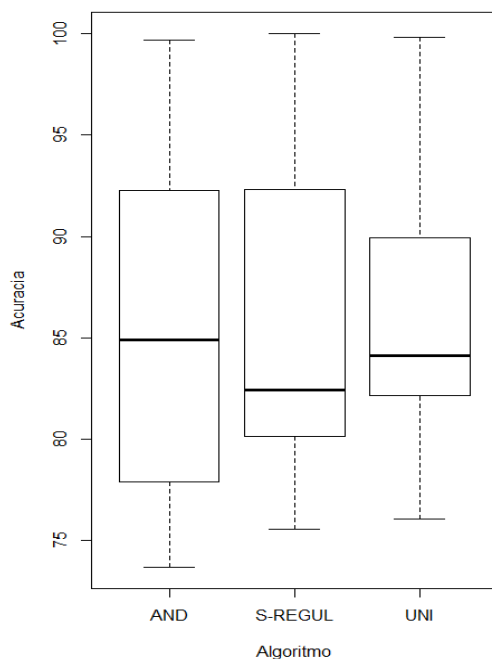
Tabela 5- Resultados das acurácias dos testes executados na comparação entre o modelo regularizado x não regularizado.

Base	F. P.	R-UNINEURON	SEM REGULARIZ.	R-ANDNEURON
FCL	5	95,73 (2,67)	92,31 (2,51)	96,13 (1,12)
HAB	3	76,37 (5,89)	75,54 (6,33)	73,69 (4,13)
IRI	2	99,82 (0,14)	100 (0,00)	99,68 (0,47)
TRA	4	82,16 (5,17)	80,15 (4,66)	77,89 (4,50)
MAM	3	84,12 (0,65)	82,43 (2,51)	84,89 (0,54)
LIV	2	76,05 (1,70)	78,33 (2,52)	74,84 (1,98)
DIA	2	83,17 (2,59)	81,57 (2,95)	80,29 (2,14)
HEA	2	88,39 (2,62)	87,10 (2,33)	87,75 (2,05)
CLI	2	89,92 (2,54)	95,11 (1,46)	92,28 (1,58)

Nas nove bases avaliadas para esses testes verifica-se que o R-UNINEURON apresentou melhores resultados de acurácia em quatro bases e o R-ANDNEURON apresentou-se como o melhor classificador em duas delas. O modelo não regularizado apresentou-se mais eficiente em três classes, porém com resultados muito próximos dos modelos regularizados. Buscando um resultado estatisticamente significativo vamos repetir

o teste de análise de variância (ANOVA) com fator de blocagem sendo cada uma das bases analisadas gerando um total de 27 grupos (3 algoritmos e 9 bases) além do valor de $\alpha=0.05$.

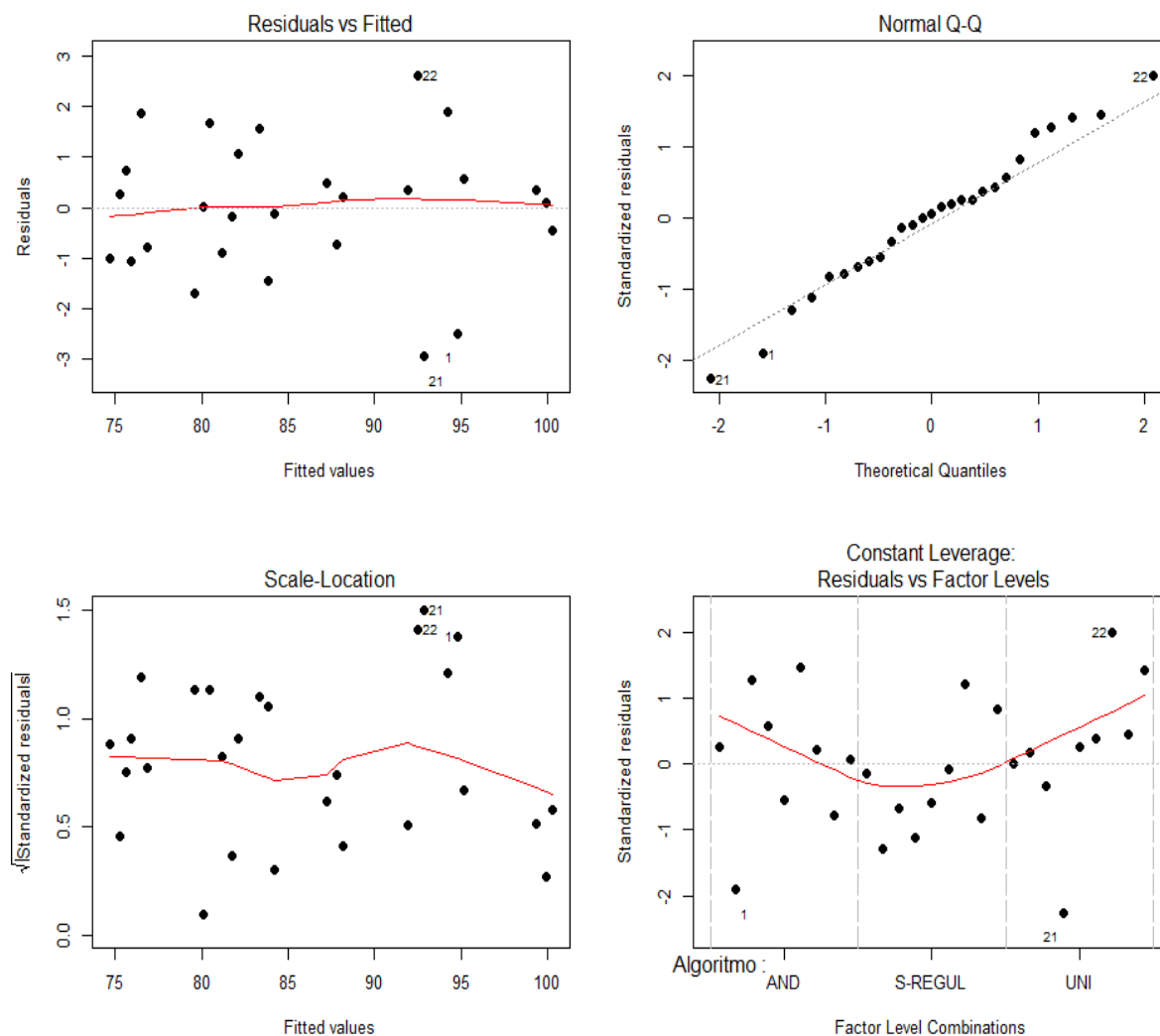
A Figura 26 apresenta o plot dos resultados dos três algoritmos analisados em relação a suas acurácias encontradas nos testes.



Fonte: Dados da própria Pesquisa

Figura 26- Plot do Teste 3 - Acurácia dos algoritmos propostos.

Após a execução da análise de variância concluiu-se que a hipótese nula deve ser aceita (p -value de 0.5210) de que o desempenho de acurácia dos três algoritmos é igual. Após a validação do modelo onde confirmamos a normalidade dos dados coletados (p -value de 0.9496), a homoscedasticidade (p -value de 0.9088) e a independência dos dados com o p -value de 0.1335 pode-se afirmar com 95 % de certeza que os três algoritmos analisados tem o mesmo comportamento ao realizar a classificação binária das nove bases avaliadas. A Figura 27 comprova as validações do teste executado.



Fonte: Dados da própria Pesquisa

Figura 27- Validações do Teste ANOVA- Teste 3.

Apesar do comportamento dos algoritmos serem iguais, podemos verificar quem teve o melhor desempenho dentre os modelos propostos. Para isso pode-se utilizar comparações *post-hoc* onde todos os desempenhos serão avaliados em pares, onde se tem uma melhor avaliação individual de cada classificador. Para realizar o teste de Dunnett (Hsu, 1996) evidenciamos o algoritmo que gostaríamos de avaliar como sendo o sem regularização. Após a execução do teste verificamos que o R-ANDNEURON apresenta no geral um comportamento pior que o SEM-REGULARIZAÇÃO e em contra partida o R-UNINEURON supera os resultados do modelo sem regularização. A Tabela 6 apresenta o resultado do teste de *Dunnett* que comprova que os algoritmos regularizados podem trazer benefícios também a acurácia geral do modelo de treinamento proposto.

Tabela 6- Resultado do Teste Post –Hoc de Dunnett.

```

Simultaneous Tests for General Linear Hypotheses

Multiple Comparisons of Means: Dunnett Contrasts

Fit: aov (formula = Acuracia ~ Algoritmo + Problema, data
= data)

Linear Hypotheses:
      Estimate Std. Error t value Pr(>|t|)
AND - S-REGUL == 0  -0.5667    0.7978  -0.710   0.705
UNI - S-REGUL == 0   0.3544    0.7978   0.444   0.868
(Adjusted p values reported -- single-step method)

```

5 Conclusões

Esse trabalho apresentou uma nova metodologia de aprendizado para as redes neurais nebulosas baseadas nos conceitos de máquina de aprendizado extremo e sobre métodos de regularização. A criação de funções de pertinência igualmente espaçadas e a definição das regras, que representam os neurônios adaptados de regras Se-Então auxiliam na interpretabilidade do processo de classificação. A forma com que os pesos da segunda camada são gerados colabora com a diminuição da complexidade e do tempo de execução do modelo. O método de regularização *bootstrap lasso* definindo a topologia mais compacta da rede auxilia também na diminuição de complexidade do modelo, permitindo que a rede com um número menor de neurônios atue de forma tão eficiente quanto os classificadores utilizados nos experimentos.

Com a metodologia de partição do espaço de entrada do modelo utilizando funções de pertinência igualmente espaçadas podemos atribuir interpretabilidade as ações do modelo de redes neurais nebulosas. O experimento que buscava esse fim deixou claro que a rede é capaz de utilizar regras interpretáveis para encontrar os dados no espaço de entrada e classificá-los de forma correta.

Os experimentos realizados e os resultados obtidos sugerem que a rede é capaz de atuar como um classificador, apresentando resultados consistentes na classificação de padrões. Nos experimentos realizados podemos comprovar que os resultados do modelo proposto são semelhantes a outros classificadores, inclusive o que se baseia em regularização.

Verificou-se que o método de regularização utilizado nesse trabalho não prejudicou a acurácia do modelo, quando comparado a um classificador que não utiliza a regularização.

Estudos para investigar outros modelos regularizadores, principalmente os que possuem conceitos ligados à otimização multiobjetivo podem ser alvo de investigações futuras.

Novos tipos de funções de pertinência podem ser adaptados para a divisão do espaço de entrada buscando a redução da complexidade na geração de regras do tipo Se Então.

Ações futuras podem ser tomadas para que o modelo seja adaptado para realizar regressão, previsão de séries temporais e classificação multi-classe.

Referências

ALLEN, Michael Patrick. **Understanding regression analysis**. Springer, 1997.

ANGELOV, Plamen. EVOLVING TAKAGI-SUGENO FUZZY SYSTEMS FROM STREAMING DATA (eTSp). **Evolving intelligent systems: methodology and applications**, v. 12, p. 21, 2010.

ANGELOV, Plamen; YAGER, Ronald. A new type of simplified fuzzy rule-based system. **International Journal of General Systems**, v. 41, n. 2, p. 163-185, 2012.

ASH, R. B. Information Theory. 1965. **Interscience, New York**.

BACH, Francis R. Bolasso: model consistent lasso estimation through the bootstrap. In: **Proceedings of the 25th international conference on Machine learning**. ACM, 2008. p. 33-40.

BACHE, K.; LICHMAN, M. UCI Machine Learning Repository. Irvine, CA: University of California, **School of Information and Computer Science**; 2013. Acesso em fevereiro de 2015.

BALLINI, R.; SOARES, S.; ANDRADE, Marinho G. Previsão de vazões médias mensais usando redes neurais nebulosas. **Sba: Controle & Automação Sociedade Brasileira de Automática**, v. 14, n. 3, p. 680-693, 2003.

BATTITI, Roberto. Using mutual information for selecting features in supervised neural net learning. **Neural Networks, IEEE Transactions on**, v. 5, n. 4, p. 537-550, 1994.

BELKIN, Mikhail; NIYOGI, Partha; SINDHWANI, Vikas. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. **The Journal of Machine Learning Research**, v. 7, p. 2399-2434, 2006.

BEZDEK, James C. **Pattern recognition with fuzzy objective function algorithms**. Kluwer Academic Publishers, 1981.

BI, Jinbo et al. Dimensionality reduction via sparse support vector machines. **The Journal of Machine Learning Research**, v. 3, p. 1229-1243, 2003.

BORDIGNON, Fernando; GOMIDE, Fernando. Capacidade de Aproximação de Redes Neurais Nebulosas Baseadas em Uninormas.

BORDIGNON, Fernando; GOMIDE, Fernando. Extreme learning for evolving hybrid neural networks. In: **Neural Networks (SBRN), 2012 Brazilian Symposium on. IEEE**, 2012. p. 196-201.

BRAGA, A. de P.; CARVALHO, ACPLF; LUDERMIR, Teresa Bernarda. Redes neurais artificiais: teoria e aplicações. **Livros Técnicos e Científicos**, 2000.

CALVO, Rodrigo. Arquitetura híbrida inteligente para navegação autônoma de robôs. 2007. **Tese de Doutorado**. Universidade de São Paulo.

CAMINHAS, Walmir M. et al. Fuzzy Set Based Neural Networks: Structure, Learning and Application. **JACIII**, v. 3, n. 3, p. 151-157, 1999.

CARRANO, Eduardo G. et al. A genetic algorithm for multiobjective training of ANFIS fuzzy networks. In: **Evolutionary Computation, 2008. CEC 2008.(IEEE World Congress on Computational Intelligence). IEEE Congress on**. IEEE, 2008. p. 3259-3265.

CYBENKO, George. Approximation by superpositions of a sigmoidal function. **Mathematics of control, signals and systems**, v. 2, n. 4, p. 303-314, 1989.

DA SILVA, Alisson Marques et al. Modelo Nebuloso Evolutivo com Seleção Adaptativa de Entradas. (2012) **II CBSF - II Congresso Brasileiro de Sistemas Fuzzy**, 933-948.

DE ALBUQUERQUE TEIXEIRA, Roselito et al. Improving generalization of MLPs with multi-objective optimization. **Neurocomputing**, v. 35, n. 1, p. 189-194, 2000.

DEB, Kalyanmoy et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. **Evolutionary Computation, IEEE Transactions on**, v. 6, n. 2, p. 182-197, 2002.

DUNN, Joseph C. A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters. 1973.

ESCANDELL-MONTERO, Pablo et al. Regularized committee of extreme learning machines for regression problems. In: **Proceedings of European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning, Special Session on Machine Ensembles**. 2012. p. 251-256.

EFRON, Bradley; TIBSHIRANI, Robert J. **An introduction to the bootstrap**. CRC press, 1994.

EFRON, Bradley et al. Least angle regression. **The Annals of statistics**, v. 32, n. 2, p. 407-499, 2004.

F.V. Berghen. LARS library: Least angled regression stagewise library. Technical report, IRIDIA, **Universit Libre de Bruxelles**, November 2005.

GUO, Lu; HAO, Jing-hua; LIU, Min. An incremental extreme learning machine for online sequential learning problems. **Neurocomputing**, v. 128, p. 50-58, 2014.

HASTIE, Trevor et al. The elements of statistical learning: data mining, inference and prediction. **The Mathematical Intelligencer**, v. 27, n. 2, p. 83-85, 2005.

HAYKIN, Simon. Neural networks: a comprehensive foundation. **Prentice Hall PTR**, 1994.

HAYKIN, Simon S. Redes neurais. **Bookman**, 2001.

HELL, Michel Bortolini. Abordagem neurofuzzy para modelagem de sistemas dinâmicos não lineares. 2008. **Tese de Doutorado**. Universidade de Campinas.

HO, Tin Kam; KLEINBERG, Eugene M. Building projectable classifiers of arbitrary complexity. In: **Pattern Recognition, 1996., Proceedings of the 13th International Conference on**. IEEE, 1996. p. 880-885.

HOERL, Arthur E.; KENNARD, Robert W. Ridge regression: Biased estimation for nonorthogonal problems. **Technometrics**, v. 12, n. 1, p. 55-67, 1970.

HOERL, Arthur E.; KENNARD, Robert W. Ridge regression: applications to nonorthogonal problems. **Technometrics**, v. 12, n. 1, p. 69-82, 1970.

HUANG, Gao et al. Semi-supervised and unsupervised extreme learning machines. **IEEE Trans Cybern.** 2014.

HUANG, Guang-Bin; ZHU, Qin-Yu; SIEW, Chee-Kheong. Extreme learning machine: theory and applications. **Neurocomputing**, v. 70, n. 1, p. 489-501, 2006.

HUYNH, Hieu Trung; WON, Yonggwan. Regularized online sequential learning algorithm for single-hidden layer feedforward neural networks. **Pattern Recognition Letters**, v. 32, n. 14, p. 1930-1935, 2011.

HSU, Chih-Wei et al. A practical guide to support vector classification. 2003.

HSU, Jason. **Multiple comparisons: theory and methods**. CRC Press, 1996.

JANG, J.-SR. ANFIS: adaptive-network-based fuzzy inference system. **Systems, Man and Cybernetics, IEEE Transactions on**, v. 23, n. 3, p. 665-685, 1993.

KIM, Jaesoo; KASABOV, Nikola. Hybrid neuro-fuzzy inference systems and their application for on-line adaptive learning of nonlinear dynamical systems. 1999.

KOHAVI, Ron et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: **Ijcai**. 1995. p. 1137-1145.

LEITE, Daniel; COSTA, P.; GOMIDE, Fernando. Evolving granular neural network for semi-supervised data stream classification. In: **Neural Networks (IJCNN), The 2010 International Joint Conference on. IEEE, 2010**. p. 1-8.

LEMOS, Andre Paim; CAMINHAS, Walmir; GOMIDE, Fernando. A fast learning algorithm for uninorm-based fuzzy neural networks. In: **Fuzzy Information Processing Society (NAFIPS), 2012 Annual Meeting of the North American. IEEE, 2012**. p. 1-6.

LEMOS, Andre; CAMINHAS, Walmir; GOMIDE, Fernando. New uninorm-based neuron model and fuzzy neural networks. In: **Fuzzy Information Processing Society (NAFIPS), 2010 Annual Meeting of the North American. IEEE, 2010.** p. 1-6.

LIMA, Elton; GOMIDE, Fernando; BALLINI, Rosangela. Participatory evolving fuzzy modeling. In: **Evolving Fuzzy Systems, 2006 International Symposium on.** IEEE, 2006. p. 36-41.

LIU, Huan; SETIONO, Rudy. Chi2: Feature Selection and Discretization of Numeric Attributes. In: **Proceedings of the Seventh International Conference on Tools with Artificial Intelligence.** IEEE Computer Society, 1995. p. 88.

LUNA, Ivette; BALLINI, Rosangela; GOMIDE, Fernando Antonio Campos. Incorporação de recorrência em estruturas neurais nebulosas. **Learning and NonLinear Models**, v. 2, n. 1, p. 49, 2004.

MACQUEEN, James et al. Some methods for classification and analysis of multivariate observations. In: **Proceedings of the fifth Berkeley symposium on mathematical statistics and probability.** 1967. p. 281-297.

MARTÍNEZ-MARTÍNEZ, José M. et al. Regularized extreme learning machine for regression problems. **Neurocomputing**, v. 74, n. 17, p. 3716-3721, 2011.

MICHE, Yoan et al. OP-ELM: optimally pruned extreme learning machine. **Neural Networks, IEEE Transactions on**, v. 21, n. 1, p. 158-162, 2010.

MITRAKIS, Nikolaos E.; THEOCHARIS, John B.; PETRIDIS, Vassilios. A multilayered neuro-fuzzy classifier with self-organizing properties. **Fuzzy sets and systems**, v. 159, n. 23, p. 3132-3159, 2008.

MURPHY, Kevin P. **Machine learning: a probabilistic perspective**. MIT press, 2012.

NEUMAIER, Arnold. Solving ill-conditioned and singular linear systems: A tutorial on regularization. **Siam Review**, v. 40, n. 3, p. 636-666, 1998.

NOCK, Richard; NIELSEN, Frank. On weighting clustering. **Pattern Analysis and Machine Intelligence, IEEE Transactions on**, v. 28, n. 8, p. 1223-1235, 2006.

PEDRYCZ, Witold. Fuzzy neural networks and neurocomputations. **Fuzzy Sets and Systems**, v. 56, n. 1, p. 1-28, 1993.

PEDRYCZ, Witold. Neurocomputations in relational systems. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 13, n. 3, p. 289-297, 1991.

PENG, Yong et al. Discriminative Graph Regularized Extreme Learning Machine for Face Recognition. **Neurocomputing** 2013.

PENROSE, Roger. A generalized inverse for matrices. In: **Mathematical proceedings of the Cambridge philosophical society**. Cambridge University Press, 1955. p. 406-413.

PORWAL, Alok; CARRANZA, E. J. M.; HALE, M. A hybrid neuro-fuzzy model for mineral potential mapping. **Mathematical geology**, v. 36, n. 7, p. 803-826, 2004.

RAMOS, Ricardo M. et al. The real-biased multiobjective genetic algorithm and its application to the design of wire antennas. **Magnetics, IEEE Transactions on**, v. 39, n. 3, p. 1329-1332, 2003.

RONG, Hai-Jun et al. A fast pruned-extreme learning machine for classification problem. **Neurocomputing**, v. 72, n. 1, p. 359-366, 2008.

ROSA, Raul Arthur Fernandes; GOMIDE, Fernando Antônio Campos. Previsão de Séries Temporais Utilizando uma Rede Neurofuzzy Evolutiva Baseada em Unineurons. **sistema**, v. 7, n. 8, p. 9.

RUMELHART, David E.; HINTON, Geoffrey E.; WILLIAMS, Ronald J. Learning representations by back-propagating errors. **MIT Press**, Cambridge, MA, USA, 1988.

SIMILÄ, Timo; TIKKA, Jarkko. Multiresponse sparse regression with application to multidimensional scaling. In: **Artificial Neural Networks: Formal Models and Their Applications–ICANN 2005**. Springer Berlin Heidelberg, 2005. p. 97-102.

SHOR, Naum Z. Cut-off method with space extension in convex programming problems. **Cybernetics and systems analysis**, v. 13, n. 1, p. 94-96, 1977.

TAKAHASHI, R. H. C.; PERES, P. L. D.; FERREIRA, P. A. V. H₂/h-infinity multiobjective pid design. **IEEE Control Systems Magazine**, v. 17, n. 5, p. 37-47, 1997.

TIKHONOV, Andrey. Solution of incorrectly formulated problems and the regularization method. In: **Soviet Math. Dokl.** 1963. p. 1035/1038.

TOU, J. T., Gonzalez, R. C. Pattern Recognition Principles. **Addison-Wesley Publishing Company**, Massachusetts, 1981.

VIRA, Chankong; HAIMES, Yacov Y. **Multiobjective decision making: theory and methodology**. North-Holland, 1983.

WAINWRIGHT, M. J. Sharp thresholds for noisy and high-dimensional recovery of sparsity using ℓ_1 -constrained quadratic programming (lasso). **IEEE Transactions on Information Theory**, v. 55, n. 5, p. 2183-2202, 2009.

YAGER, Ronald R.; RYBALOV, Alexander. Uninorm aggregation operators. **Fuzzy sets and systems**, v. 80, n. 1, p. 111-120, 1996.

YUAN, Ming; LIN, Yi. On the non-negative garrotte estimator. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, v. 69, n. 2, p. 143-161, 2007.

ZADEH, Lotfi A. Fuzzy sets. **Information and control**, v. 8, n. 3, p. 338-353, 1965.

ZHAO, Jianwei; WANG, Zhihui; PARK, Dong Sun. Online sequential extreme learning machine with forgetting mechanism. **Neurocomputing**, v. 87, p. 79-89, 2012.

ZHAO, Peng; YU, Bin. On model selection consistency of Lasso. **The Journal of Machine Learning Research**, v. 7, p. 2541-2563, 2006.

ZHOU, Xinran; LIU, Zijian; ZHU, Congxu. Online Regularized and Kernelized Extreme Learning Machines with Forgetting Mechanism. **Mathematical Problems in Engineering**, v. 2014, 2014.

ZOU, Hui. The adaptive lasso and its oracle properties. **Journal of the American statistical association**, v. 101, n. 476, p. 1418-1429, 2006.

ZOU, Hui; HASTIE, Trevor. Regularization and variable selection via the elastic net. **Journal of the Royal Statistical Society: Series B (Statistical Methodology)**, v. 67, n. 2, p. 301-320, 2005.