



Classificador Neuro-nebuloso de Desempenho Escolar Usando o ANFIS com Implementação em R

Jeferson Costa da Silva¹, Miguel Raymundo Flores Santibañez¹

¹ Departamento de Ciência da Computação – Universidade Federal de Roraima (UFRR)
Cep 69310-000 – Boa Vista – RR – Brazil

jeferson.cosil@{hotmail,gmail}.com,

miguel.santibanez@ufrr.br

Abstract. This paper describes some fundamental concepts for the implementation of a neuro-fuzzy classifier school performance. Sort of school performance information is a non-linear task because it involves values (“Attributes”) as: note, percentage of attendance (punctuality) and disapproval of students in a course. The objective is to qualify the performance of students in fuzzy concepts (“Fuzzy Classes”) as poor, bad, regular, good or great. Based on numerical information contained in a dataset with the attribute values is developed a fuzzy classifier based on the model of ANFIS learning and neuro-fuzzy classification and implemented in the statistical programming language R.

Resumo. Este artigo descreve alguns conceitos fundamentais para a implementação de um classificador neuro-nebuloso de desempenho escolar. Classificar informações de desempenho escolar é uma tarefa não-linear, pois envolve valores (“Atributos”) como: nota, porcentagem de assiduidade (pontualidade) e número de reprovações dos alunos em uma disciplina. O objetivo é qualificar o desempenho dos alunos em conceitos nebulosos (“Classes Nebulosas”) como: péssimo, ruim, regular, bom ou ótimo. Baseado em informações numéricas contidas em um dataset com os valores dos atributos é desenvolvido um classificador nebuloso baseado no modelo de ANFIS de aprendizagem e classificação neuro-nebulosa e implementado na linguagem de programação estatística R.

1. Introdução

Segundo [Zadeh 1965] a Lógica Nebulosa (do inglês Fuzzy Logic) é o raciocínio matemático que permite aproximar o modo de pensar dos seres humanos ao raciocínio exato. O pensamento humano é bastante complexo, redundante, abstrato, etc. Por este motivo sua implementação diretamente no hardware ou software dos computadores é ineficiente. Utilizando o raciocínio lógico da teoria dos conjuntos numéricos para solucionar tal problema [Zadeh 1973] propôs que ao invés de utilizar os valores lógicos precisos, como: TRUE ou 1 (para representar uma proposição verdadeira) e FALSE ou 0 (para representar uma proposição falsa); utilizou valores “intermediários” entre o 1 e o 0 para representar a forma do ser humano de pensar.

O problema que será abordado neste artigo é o de classificar informações imprecisas e vagas por meio de uma rede neural híbrida, ou seja, que combina o



poder de aproximação das redes neurais com heurísticas baseadas em regras de raciocínio fuzzy. Para isso será implementada uma rede

Para [Gomide et al 1995] a Lógica Nebulosa modela formas aproximadas de raciocínio de uma expressão enquanto a lógica proposicional e de predicados tem formas de raciocínio muito bem definidas e precisas. A vantagem em empregar esta técnica é a capacidade de representar expressões e raciocínios imprecisos e vagos através do grau de pertinência/relacionamento de um valor a um determinado conjunto numérico discreto, que variam entre 0 e 1, ou seja, $[0, 1]$. O raciocínio humano lida melhor quando as informações de quantificação são tratadas de maneira vaga. Por esse motivo a lógica nebulosa ajuda a desenvolver sistemas computacionais capazes de aprender a qualificar informações vagas por meio das Redes Neurais Artificiais.

Por exemplo: se definirmos algumas faixas de temperatura como sendo conjuntos, entre, $0^{\circ}C$ e $25^{\circ}C$ são consideradas temperaturas baixas, por tanto, conjunto nebuloso temperatura baixa; entre $26^{\circ}C$ e $75^{\circ}C$ consideram-se temperaturas médias - conjunto nebuloso temperatura média; e maior que $75^{\circ}C$ temperaturas altas - conjunto nebuloso de temperatura alta. Ao fazer uma leitura de $20^{\circ}C$, observa-se que esse valor tem uma maior pertinência com o conjunto nebuloso baixa que com os demais conjuntos nebulosos, ao passo que, se o valor aferido por um termômetro indicar $50^{\circ}C$, esse (valor) terá maior relacionamento com o conjunto nebuloso média. Os conjuntos nebulosos abordados nesse exemplo são chamados de variáveis linguísticas, pois expressam diretamente uma ideia de quantificação imprecisa.

Para [Gomide et al 1995] a definição formal de conjunto nebuloso é uma coleção U de objetos denominados genericamente por $\{u\}$. U é chamado de universo de discurso podendo ser contínuo ou discreto. Um conjunto nebuloso A em um universo de discurso U é definido por uma função de pertinência μ_A que assume valores em um intervalo $[0, 1]$: $\mu_A : U \rightarrow [0, 1]$. O conjunto nebuloso A em U é, então um conjunto de pares ordenados $A = \{\mu_A(u)/u\}$, $u \in U$.

Combinando conceitos de Redes Neurais Artificiais com a Lógica Nebulosa, permite-se o projeto e a implementação de sistemas de classificação e regressão neuro-nebulosos com capacidade de aprendizagem de máquina. Para [Lin e Lee 1996], o primeiro pesquisador a propor o pensamento nebuloso foi o professor Lotfi Asker Zadeh da Universidade da Califórnia em Berkeley no ano de 1965, segundo o qual a lógica nebulosa foi desenvolvida para tratar o aspecto vago e incerto da informação.

Os primeiros trabalhos a serem publicados sobre a aprendizagem artificial foram escritos por [McCulloch e Pitts 1943], que simulavam o comportamento do neurônio biológico através da interconexão de componentes eletrônicos, como válvulas (transistores), resistores e capacitores. Por serem os primeiros trabalhos nesta área o modelo de neurônio artificial desenvolvido por eles ficou conhecido como neurônio de McCulloch-Pitts (M-P), no qual simulavam as conexões sinápticas do cérebro através da mudança de valores de resistores e da corrente elétrica em amplificadores eletrônicos.

O objetivo deste artigo é mostrar uma forma de implementar, treinar e testar



uma rede neuro-nebulosa para classificação de dados, contidos em um dataset, na linguagem R. O modelo neuro-nebuloso baseia-se nos conceitos de Redes Neurais Artificiais (aprendizagem supervisionada de máquina) e da Lógica Nebulosa (informações imprecisas) para classificação de informações contidas em bases de dados de teste dataset, em outras palavras, é implementado um tipo de Rede Neural Artificial que aprende a classificar informações utilizando heurísticas nebulosas para ajustes do pesos sinápticos da rede.

Sistemas de classificação utilizando Redes Neurais Artificiais, como por exemplo: o ADALINE e o MADALINE [Widrow 1987], foram criadas pensando numa maneira de uma RNA aprender a solucionar problemas relacionados a classificação/separação de informações de funções lógicas, por tanto, utiliza-se de informações precisas para a aprendizagem de máquina. O modelo de classificador abordado neste artigo é mais amplo, pois classifica informações nebulosas contidas em dataset ou bases de dados para auxiliar na tomada de decisão e na aprendizagem de máquina. Segundo [Abraham 2004], os sistemas computacionais neuro-fuzzy são frameworks populares para solucionar problemas complexos envolvendo conhecimento expresso em bases de regras linguísticas.

A linguagem R para implementação do modelo neuro-nebuloso ANFIS foi escolhida por ser uma ferramenta free, cujo uso tem crescido muito nos últimos anos, conter pacotes para implementação de modelos de aprendizagem neuro-nebulosa estáveis com suporte e haver poucas pesquisas com implementação nesta linguagem.

Por fim, outra vantagem em utilizar os pacotes da linguagem R é o fato de existir uma vasta documentação disponível para consulta e suporte aos erros apresentados pela comunidade de pesquisadores em inteligência de artificial, ou seja, a linguagem R tornou-se uma alternativa, viável, a ferramentas proprietárias, como o MATLAB, pois apresenta resultados semelhantes no desenvolvimento e implementação de sistemas de aprendizagem de máquina.

2. Raciocínio Nebuloso

Nesta seção será abordado o principal método e técnica para classificação usando raciocínio nebuloso, as técnicas para classificação de regras base, determinação dos conjuntos e classes que fazem parte do escopo dos sistemas de controle e classificação de nebulosos. Quando se deseja implementar um sistema de controle autônomo nebuloso é necessário estabelecer regras nebulosas, para classificar informações nebulosas.

Antes de iniciar, é importante classificar os principais métodos de raciocínio nebuloso, [Tanaka 1997] classifica o raciocínio nebuloso em dois tipos: métodos diretos e métodos indiretos, dentre os quais se destacam os métodos diretos (Método Direto de [Mamdani 1977] e Método de Modelagem Nebulosa de Takagi-Sugeno) ou o modelo aprimorado Takagi-Sugeno-Kang TSK. O modelo nebuloso abordado neste artigo é o de Takagi-Sugeno-Kang TSK que é utilizado para implementar o Sistema de Inferência Neuro-Nebulosa Adaptativa-ANFIS.

3. Método Mamdani

3.1. Nebulização (Fuzzifier)

A função da etapa de nebulização é fazer uma avaliação subjetiva de um valor de entrada previamente mensurado e o mapeia para uma avaliação subjetiva. [Lin e Lee 1996] afirmam que essa etapa pode ser definida como um mapeamento de um espaço de entrada observando os rótulos de conjuntos nebulosos num universo de entrada especificado no discurso. Uma vez que a manipulação de dados em um controlador lógico nebuloso baseado na teoria dos conjuntos nebulosos, a nebulização é necessária e desejável em um estágio inicial. Em outras palavras, a etapa de nebulização converte os valores reais de entrada em valores nebulosos, fazem o processo escalar \rightarrow nebuloso.

A operação de nebulização natural é simples e consiste basicamente em converter um valor x_0 Crisp-(com valores binários) em um singleton nebuloso A - é um conjunto nebuloso cuja função de pertinência é igual a unidade em um ponto particular e zero no restante - dentro de um universo especificado de discurso, isto é, a transformação dos valores das variáveis de entrada do problema em valores linguísticos nebulosos.

De acordo com [Almeida et al 2003], o processo de conversão de um valor real de entrada, ou seja, no processo escalar \rightarrow nebuloso, os antecedentes de cada regra são processados através da interseção nebulosa entre os graus de relacionamento das entradas atuais nos termos primários definidos em cada uma. Este processo gera um grau de relacionamento de disparo para cada regra de controle. Sendo determinado pelo cálculo de um coeficiente de disparo ou de ativação $D^{(k)}$ (Equação 1) da k -ésima regra de controle nebulosa, onde os k índices nos conjuntos nebulosos denotam os termos primários que compõem a regra k na base de inferência. Este processamento transforma informações quantitativas em informações qualitativas, sendo considerado um processo de generalização.

$$D^{(k)} = T[\mu_{A_1^k}(x_1), \mu_{A_2^k}(x_2), \dots, \mu_{A_p^k}(x_p)] = \min[\mu_{A_1^k}(x_1), \mu_{A_2^k}(x_2), \dots, \mu_{A_p^k}(x_p)] \quad (1)$$

Todas as regras para as quais o coeficiente de disparo for maior que zero são ditas regras que dispararam para as entradas atuais. Isto quer dizer que elas vão contribuir para o cálculo da saída correspondente do sistema de inferência. Estes coeficientes de disparo, por sua vez, vão limitar os valores máximos dos conjuntos nebulosos de saída gerados por estas regras. Finalmente, uma operação global de união vai compor um conjunto nebuloso para cada variável de saída, contendo informações sobre todas as regras disparadas para as entradas atuais. A Equação 2 mostra a composição de um conjunto nebuloso para o caso de uma saída de regra mostrada na Equação 3.

$$\mu_{C_i'}(z) = S_{k=1 \dots n}[\top(D^{(k)}, \mu_{C_i}(z))] = \max_{k=1 \dots n}[\min(D^{(k)}, \mu_{C_i}(z))], \forall z \in C. \quad (2)$$

O processo de inferência descrito transforma uma informação qualitativa em outra informação qualitativa, através de uma conversão. Para [Almeida et al 2003] o conjunto nebuloso gerado durante o processo de inferência pode então ser utilizado

diretamente em um diagnóstico qualitativo de tomada de decisão, ou vai ser convertido em um valor escalar proporcional para atuação externa através de atuadores convencionais, no caso de um sistema de controle.

3.2. Base de Regras Nebulosas (Fuzzy Rule Base)

Para [Ross 2010] as regras de controle nebuloso são caracterizadas por uma coleção de instruções SE-ENTÃO ou (IF-THEN), nas quais os antecedentes e conseqüências envolvem as variáveis linguísticas do sistema de controle. Esta coleção de regras de controle nebuloso ou declarações nebulosas caracteriza a relação de entrada/saída (I/O) simples do sistema nebuloso.

A forma geral para especificação de regras de controle nebuloso, para sistemas de múltiplas entradas e saída simples, é descrita pelo modelo da Equação 3:

$$R^i = \text{SE } x \text{ é } A_i, \dots, E \ y \text{ é } B_i, \text{ ENTÃO } z = C_i, i = 1, 2, 3, \dots, n \quad (3)$$

Onde x, \dots, y e z são variáveis linguísticas que representam o estado do sistema de controle, A_1, \dots, B_1 e C_1 são valores linguísticos das variáveis x, \dots, y e z nos universos de discurso U, \dots, V e W . A regra de controle da Equação 3 avalia o estado do processo em tempo t , computa e decide o controle das ações em função do estado das variáveis x, \dots, y e z .

3.3. Motor de Inferência (Engine Inference)

Para [Kaufmann e Gupta 1991] este é o kernel do sistema de controle nebuloso, pois modela a tomada de decisões dentro do quadro conceitual da lógica nebulosa e do raciocínio aproximado ao dos seres humanos. A generalização modus ponens desempenha um papel importante neste contexto, para aplicar ao raciocínio nebuloso em sistemas de controle, a generalização pode ser rescrita da seguinte maneira:

$$\begin{array}{l} \text{Entrada : } x \text{ é } A' \text{ E } y \text{ é } B' \\ \text{Premissa 1 : SE } x \text{ é } A, \text{ ENTÃO } y \text{ é } B \\ \text{Premissa 2 : } x \text{ é } A' \\ \hline \text{Conclusão : } y \text{ é } B' \end{array}$$

Onde A, A', B e B' são predicados no conjunto universal U e V respectivamente. Em geral, uma regra de controle nebulosa é uma relação quando é expressa como uma implicação nebulosa $R = A \rightarrow B$.

3.4. Desnebulização (Defuzzifier)

Para [Lin e Lee 1996] a etapa de desnebulização é um mapeamento do espaço de ações do controlador nebuloso definido sobre uma saída de um universo de discurso dentro de um espaço de controle não nebuloso. Este processo é necessário porque muitas aplicações práticas requerem ações de controle Crisp para acionar o controle. Assim a desnebulização é necessária quando o raciocínio nebuloso é usado.

A estratégia de desnebulização visa produzir uma ação de controle não nebuloso que melhor represente a distribuição das possibilidades de uma ação de controle nebuloso inferido. Não existe um procedimento sistemático para a escolha de uma

estratégia de desnebulização, porém comumente são utilizados dois métodos para desnebulizar um valor inferido, que são: o método do centro da área COA e método de média máxima MOM.

A estratégia adotada pelo método do centro da área COA é calcular o valor do centro de gravidade da distribuição, no caso de um universo discreto, este método usa a Equação 4:

$$Z_{COA}^* = \frac{\sum_{j=1}^n \mu_A(z_j) \cdot z_j}{\sum_{j=1}^n \mu_A(z_j)} \quad (4)$$

Em que n é o número de níveis de quantização de saída, z_j é a quantidade de saída de controle ao nível de quantização e $\mu_A(z_j)$ representa o valor da função relacionamento na saída do conjunto nebuloso A.

A estratégia adotada pelo método de média máxima MOM é representar o valor médio de todas as ações de controle local de quem estiver ao alcance máximo da função de relacionamento, no caso de um universo discreto, as ações de controle podem ser expressas pela Equação 5:

$$Z_{MOM}^* = \frac{\sum_{j=1}^n z_j}{\sum_{j=1}^n m} \quad (5)$$

Em que z_j é o valor suporte ao qual a função relacionamento atinge o valor máximo $\mu_A(z_j)$ e m é o número de tais valores de suporte.

3.5. Método Takagi-Sugeno-Kang

Este modelo de raciocínio nebuloso foi proposto por Takagi, Sugeno e Kang-(TSK), na tentativa de desenvolver um sistema aproximado para generalização das regras para um determinado conjunto de dados de entrada ou saída. Uma regra nebulosa típica no modelo TSK tem a forma da Equação 6:

$$\text{SE } x \text{ é } A \text{ E } y \text{ é } B \text{ ENTÃO } z = f(x_1, x_2, \dots, x_p) \quad (6)$$

De acordo com [Jang et al 1997], A e B são os conjuntos nebulosos no antecedente, enquanto que $z = f(x_1, x_2, \dots, x_p)$ é uma função Crisp na consequência. Normalmente $f(x_1, x_2, \dots, x_p)$ é um polinômio das variáveis de entrada (x_1, x_2, \dots, x_p) , mas pode ser qualquer função, desde que descreva corretamente o resultado do modelo na região nebulosa especificada pelo antecedente da regra. Quando $f(x_1, x_2, \dots, x_p)$ é um polinômio de primeira ordem o resultado do sistema de inferência nebuloso é chamado de: Modelo TSK Nebuloso de primeira ordem, que foi proposto inicialmente por Takagi, Sugeno e Kang.

Quando f é constante tem-se um Modelo TSK Nebuloso de ordem zero, que pode ser visto como um caso especial do sistema de inferência Mamdani, em que cada consequência de regra é especificada por uma função relacionamento de uma função degrau de centro constante. Além disso um Modelo TSK Nebuloso de ordem zero é funcionalmente equivalente a uma rede de função de base radial sob certas restrições menores.

A saída de um modelo TSK nebuloso de ordem zero é uma função suave de suas variáveis de entrada, desde que o vizinho da função relacionamento no antecedente se sobreponha. Em outras palavras, a sobreposição das funções de relacionamento na consequência do modelo de Mamdani não tem um efeito decisivo na suavidade, é a sobreposição do antecedente da função relacionamento que determina a suavidade do comportamento da entrada ou saída.

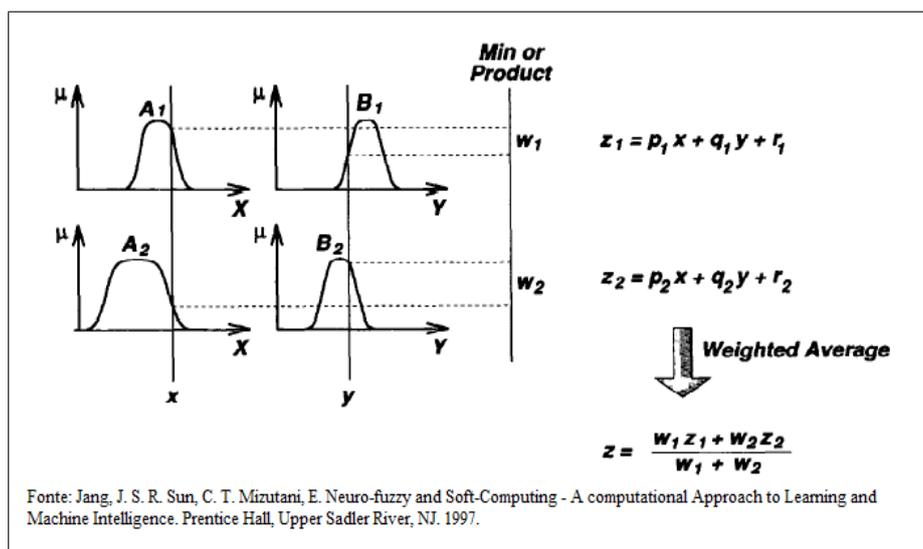


Figura 1. O Modelo TSK

A Figura 1 mostra o procedimento do raciocínio para o modelo TSK nebuloso de primeira ordem. Desde que cada regra tenha uma saída simples, a saída total é obtida por meio da média ponderada, evitando, assim, o processo demorado de desnebulização necessária em um modelo Mamdani. Na prática, o operador média ponderada às vezes é substituído por um operador de soma ponderada (que é $z = w_1.z_1 + w_2.z_2$ na figura Figura 1) para reduzir o número de computações, especialmente no treinamento de um sistema de inferência nebulosa. Contudo, esta simplificação poderia levar prejuízos aos sentidos da função de pertinência a menos que a soma dos pontos fortes de disparo (que é $\sum_i w_i$) estejam próximos à unidade. Uma vez que a única parte nebulosa de Modelo TSK é antecedente, é fácil demonstrar a distinção entre um conjunto de regras nebulosas e os não-nebulosos.

Por exemplo: Seja de um Modelo Sugeno nebuloso de duas entradas e saída simples com quatro regras expressas da seguinte forma:

$$\left\{ \begin{array}{l} \text{Se } X \text{ é pequeno e } Y \text{ é pequeno então } z = -x + y + 1 \\ \text{Se } X \text{ é pequeno e } Y \text{ é grande então } z = -y + 3 \\ \text{Se } X \text{ é grande e } Y \text{ é pequeno então } z = -x + 3 \\ \text{Se } X \text{ é grande e } Y \text{ é grande então } z = x + y + 2 \end{array} \right.$$

De acordo com [Takagi 1990], ao contrário do modelo nebuloso de Mamdani, o modelo TSK não pode seguir a regra de composição de inferência estritamente em seu mecanismo de raciocínio. Isso representa algumas dificuldades quando as entradas para o modelo TSK são nebulosas. Especificamente, pode-se ainda empregar a

correspondência de conjuntos nebulosos para encontrar a condição de ativação da regra. No entanto, o resultado geral de saída através da média ponderada ou soma ponderada é sempre Crisp. Isto é contraintuitivo uma vez que um modelo nebuloso pode ser capaz de propagar a incerteza da entrada para a saída de modo adequado.

De acordo com [Almeida et al 2003] os modelos nebulosos TSK são também baseados na utilização de uma base de regras condicionais de inferência. Mas, neste caso, as consequências das regras, ao invés de serem formados por relações nebulosas, compõem-se de equações paramétricas relacionando as entradas e as saídas do processo. A sintaxe do banco de conhecimento é diferente do caso do modelo de Mamdani, implicando em alterações também das propriedades semânticas correspondentes. A Equação 6 mostra uma regra genérica de um modelo TSK.

O processamento de conhecimento em um modelo TSK é semelhante ao que ocorre em modelos de Mamdani. A etapa de conversão escalar \rightarrow nebuloso é idêntica ao modelo mostrado na equação 1, exceto pelo fato de que a norma \top utilizada na maioria das aplicações não ser a função $\min(\cdot)$, a Equação 7 é usada para conversão de valores escalares \rightarrow nebulosos no modelo TSK.

$$D^{(k)} = \top[\mu_{A_1^k}(x_1), \mu_{A_2^k}(x_2), \dots, \mu_{A_p^k}(x_p)] \quad (7)$$

A saída numérica é calculada diretamente pela soma das saídas das regras, ponderada pelos valores de ativação $D^{(k)}$ de cada uma delas. Um modelo TSK com múltiplas saídas sempre pode ser transformado em múltiplos modelos de saída única, a Equação 8 é usada para calcular os valores da saída numérica do modelo.

$$\hat{z} = \frac{\sum_{i=1 \dots k} D^{(i)} \cdot f_i(x_1, x_2, \dots, x_p)}{\sum_{i=1 \dots k} D^{(i)}} \quad (8)$$

É comum a utilização de um modelo TSK em substituição a um modelo matemático convencional em um esquema de controle ou modelagem de sistemas reais. Para ser utilizado desta maneira, é necessário que o modelo seja ajustado de forma a se comportar como o sistema real que esta representando. Assim, devem ser criadas partições nebulosas para as variáveis de entrada, ou seja, são definidos conjuntos nebulosos nos universos de discurso destas variáveis. Através de um conjunto de dados de entrada e saída do sistema a ser modelado, os parâmetros das consequências destas regras são estimadas segundo algum índice de desempenho definido pelo usuário. A minimização do erro quadrático entre a saída do modelo TSK e os dados de saída disponíveis, normalmente utilizada como medida do desempenho. Neste caso, um algoritmo de mínimos-quadrados recursivo MQR pode ser utilizado para realizar esta estimação.

Para [Almeida et al 2003], uma vez que as partições nebulosas e os parâmetros P de saída estão otimizados segundo o índice escolhido pelo usuário, o modelo esta pronto para substituir o modelo convencional existente. Na Equação 9 a expressão que relaciona as entradas e as saídas de um modelo TSK de primeira ordem utiliza um conjunto de k regras de produção nebulosa e p entradas. Ela é consistente com

a regra de produção genérica mostrada na Equação 1.

$$z(x_1, x_2, \dots, x_n) = \sum_{i=1}^k \beta_i \cdot (P_{i0} + P_{i1} \cdot x_1 + \dots + P_{ip} \cdot x_p) \quad (9)$$

Onde

$$\beta_i = \frac{\top[\mu_{A_{i1}}(x_1), \dots, \mu_{A_{ip}}(x_p)]}{\sum_{j=1}^k \top[\mu_{A_{i1}}(x_1), \dots, \mu_{A_{ip}}(x_p)]} \quad (10)$$

A seguir, é descrito de forma resumida um método de ajuste de modelos TSK baseado em um índice de desempenho quadrático. Dado um conjunto de dados de entrada e saída representado por: $x_{1j}, x_{2j}, \dots, x_{kj}, y_j$ ($j = 1, 2, \dots, m$), o conjunto de parâmetros P das consequências das regras nebulosas pode ser estimado por MQR. Considerando uma matriz X de dimensões $m \times k \cdot (p+1)$, um vetor Y de dimensão m e um vetor P de parâmetros de dimensão $k \cdot (p+1)$ tem-se a Equação 11:

$$X = \begin{bmatrix} \beta_{11} & \dots & \beta_{k1} & x_{11} \cdot \beta_{11} & \dots & x_{11} \cdot \beta_{k1} \\ & & & x_{p1} \cdot \beta_{11} & & \\ \dots & & & \dots & & \dots \\ \beta_{1m} & & & & x_{1m} \cdot \beta_{1m} & \\ \dots & & & x_{p1} \cdot \beta_{1m} & & x_{p1} \cdot \beta_{km} \end{bmatrix} \quad (11)$$

$$Y = [y_1, \dots, y_m]^T \quad (12)$$

$$P = [P_{10}, \dots, P_{k0}, P_{11}, \dots, P_{k1}, \dots, P_{kp}]^T \quad (13)$$

Então os parâmetros P podem ser calculados por:

$$P = (X^T \cdot X)^{-1} \cdot X^T \cdot Y \quad (14)$$

Ou, alternativamente, podem ser estimados de forma recursiva através de um filtro de Kalman em regime permanente do tipo mostrado pelas Equações 15 e 16:

$$P_{i+1} = P_i + S_{i+1} \cdot x_{i+1} \cdot (y_{i+1} - x_{i+1} \cdot P_i) \quad (15)$$

$$S_{i+1} = S_i - \frac{S_i \cdot x_i + x_{i+1} \cdot S_i}{1 + x_{i+1} \cdot S_i \cdot x_{i+1}^T} \quad (i = 0, 1, \dots, m-1) \quad (16)$$

Nas Equações 15 e 16, x_i são linhas da matriz X . Os valores iniciais P_0 e S_0 são definidos nas Equações 17 e 18 a seguir, onde α é um número escalar positivo com grande módulo e I uma matriz identidade de dimensões $k \cdot (p+1) \times k \cdot (p+1)$.

$$P_0 = 0 \quad (17)$$

$$S_0 = \alpha.I \quad (18)$$

A matriz S é chamada de matriz de covariâncias. Ela mede a correlação entre os parâmetros P durante o treinamento.

4. Conceitos de Redes Neurais Artificiais

De acordo com [Neto e Nicoletti 2005] as Redes Neurais Artificiais (RNA's) são modelos matemáticos computacionais baseados no funcionamento do neurônio humano, ou seja, são algoritmos que simulam a capacidade de aprendizado nas máquinas a partir de um conjunto de dados agrupados a serem processados. As RNA's constituem um vasto campo de pesquisa, pois estão ligadas à maneira como cada pesquisa trata o funcionamento do cérebro humano, uma vez que os modelos matemáticos mais realísticos de RNA estão intimamente relacionados às pesquisas no campo da neurociência.

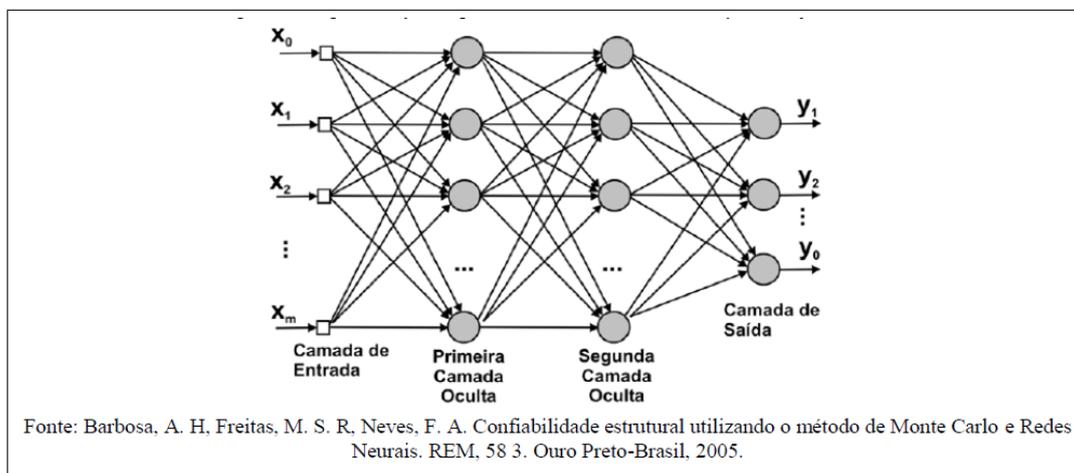


Figura 2. Representação esquemática de uma RNA *Multilayer Feedforward*.

A Figura 2 mostra uma RNA, Multilayer Feedforward (Backpropagation), com diversas camadas. A primeira camada de neurônios é chamada camada de entrada, as camadas no meio são chamadas de camadas ocultas e a última é chamada camada de saída.

Sistemas neuro-artificiais têm sido construídos para usar alguns princípios organizacionais semelhantes ao do cérebro humano. As redes neurais constituem uma nova geração de métodos e algoritmos matemáticos utilizados no processamento de dados. As RNA's são boas para processar informações do tipo: classificação/correspondência de padrões, aproximação de funções, otimização, agrupamento de dados, etc. enquanto os algoritmos tradicionais são mais eficientes para realização de operações lógicas aritméticas. A vantagem em se utilizar RNA's é a possibilidade das máquinas e aprenderem a automatizar um ambiente a ser controlado.

Para [Faceli et al 2011] as RNA's são constituídas de várias unidades básicas de processamento, chamadas de nós, unidades ou neurônios, interconectados e operando em paralelo possibilitando que as RNA's possam se reconfigurar e aprender

a processar informações, de forma autônoma. O cérebro humano é constituído de bilhões de neurônios e redes sinápticas, nas RNA's estas conexões são responsáveis por receber e processar as informações dos conjuntos de dados na entrada da rede.

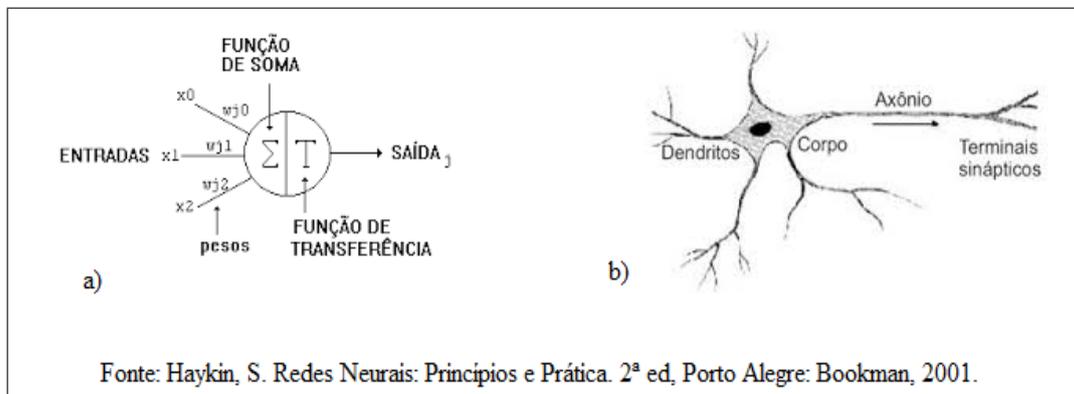


Figura 3. Correspondência entre um neurônio artificial em (a) e um neurônio biológico em (b).

A Figura 3 mostra a correspondência que existe entre um neurônio artificial, em (a), e um neurônio biológico, em (b). O neurônio biológico é ativado através de substâncias químicas que são recebidas pelo corpo do neurônio e reagem ativando as saídas, enquanto que no neurônio artificial a ativação da saída (output path) depende dos valores dos pesos sinápticos (weights) w_{ij} , do valor do parâmetro Bias (θ_i) e dos valores de entrada x_n (inputs).

Segundo [Lin e Lee 1996], a ativação de um neurônio artificial M-P, como mostrado na Figura 3(b), é realizada pelo modelo matemático representado na Função 19, porém para ser ativada a saída do neurônio o somatório do produto dos valores dos pesos sinápticos w_{ij} e dos valores de entrada x_n e subtraído do somatório o valor do Bias θ_i tem que ser maior ou igual a 0 (zero), ou seja, o resultado da Equação 20 precisa ser maior ou igual a 0 (zero) para que a saída do neurônio seja ativada, satisfazendo as condições estabelecidas na Função 19.

$$a(f) = \begin{cases} 1 & \text{se } f \geq 0 \\ 0 & \text{se } f < 0 \end{cases} \quad (19)$$

$$y_i(t+1) = a. \sum_{j=1}^m (w_{ij}.x_j(t)) - \theta_i \quad (20)$$

De acordo com [Coppin 2012], cada neurônio, em uma rede neuronal, recebe uma série de entradas x_n . Uma função chamada de função ativação é aplicada a esses valores de entrada, o que resulta no nível de ativação do neurônio, que é o valor de saída do neurônio. Existem modelos de funções que podem ser utilizadas nos neurônios, em que algumas funções são mais utilizadas para modelar os neurônios artificiais que outras.

As funções mais conhecidas e usadas para modelar a ativação da saída dos neurônios são mostradas na Figura 4(a); que apresenta o modelo da função degrau;

a Figura 4(b), o modelo da função sigmoide, e a Figura 4(c) o modelo da função linear.

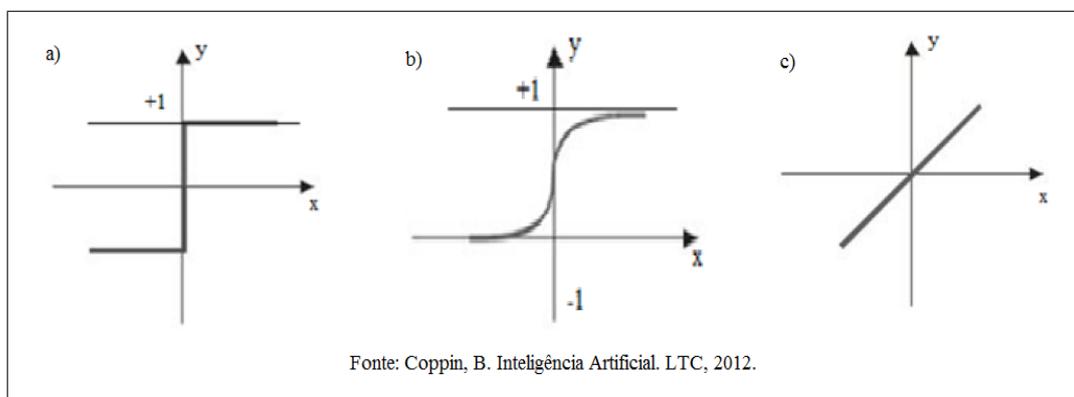


Figura 4. Funções de Ativação da saída em uma RNA a) Degrau b) Sigmoide c) Linear

As RNA's constituem métodos de processamento paralelo, bastante úteis, pois têm a capacidade de reconhecer padrões ou mesmo classificar variados tipos de dados. Para isso é necessário que, ao ser projetada, uma rede neural seja treinada para assim aprender como processar as informações nas entradas. O treinamento consiste em alterar os valores da rede a fim de minimizar o erro médio quadrático entre a saída projetada e aquela produzida pela rede neural.

Outra etapa importante é o aprendizado da rede, que consiste no ajuste dos valores dos pesos sinápticos w_{ij} e do valor do parâmetro Bias θ_i de cada neurônio que compõe a rede a fim de que a máquina processe as informações contidas no ambiente com maior precisão. Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede esta inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre. [Haykin, 2001]

5. Sistema de Inferência Neuro-Nebulosa Adaptativa-ANFIS

Segundo [Abrahan 2004] é uma classe de Rede Neural Adaptativa que é funcionalmente equivalente a um sistema de inferência nebulosa. O modelo proposto se refere como o (Adaptative Neuro-Fuzzy Inference System-ANFIS) (Sistema de Inferência Neuro-Nebulosa Adaptativa) implementa uma rede neural com aprendizado nebuloso para classificação de informações. Descreve-se como compor o conjunto de parâmetros para facilitar a aprendizagem híbrida de regras do ANFIS com o método de Takagi-Sugeno-Kang TSK. A Figura 5 mostra a representação gráfica abordada para construção de um sistema de classificação utilizando o ANFIS como classificador nebuloso com pesos normalizados.

5.1. Arquitetura ANFIS

Este método foi proposto por [Jang et al 1997], usa o raciocínio TSK na parte consequente das regras nebulosas (SE < antecedente > ENTÃO < consequente >). Utiliza regras do tipo mostrado pelas Equações 21 e 22 para classificação das

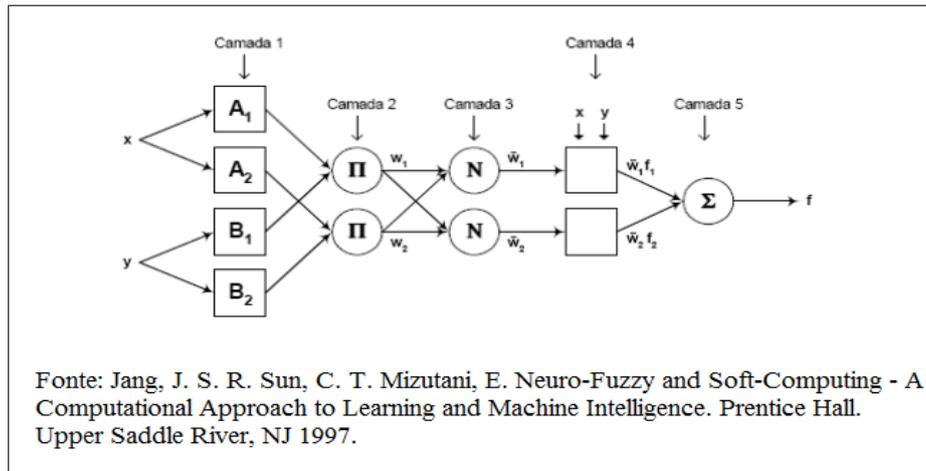


Figura 5. Arquitetura ANFIS para o modelo TSK, com pesos normalizados na última camada.

informações, onde x ou y são os valores de entradas, A ($= A_1, A_2$) ou B ($= B_1, B_2$) são conjuntos nebulosos (tais como “Notas” ou “Assiduidade”), f_i é a variável de saída (um polinômio linear de primeira ordem sobre as variáveis de entrada), $(p_i, q_i$ e $r_i)$ são os parâmetros consequentes da regras nebulosas e i é o índice que referência a regra nebulosa a qual os parâmetros pertencem.

$$\text{Regra 1: Se } x \text{ é } A_1 \text{ E } y \text{ é } B_1, \text{ ENTÃO } f_1 = p_1 \cdot x + q_1 \cdot y + r_1, \quad (21)$$

$$\text{Regra 2: Se } x \text{ é } A_2 \text{ E } y \text{ é } B_2, \text{ ENTÃO } f_2 = p_2 \cdot x + q_2 \cdot y + r_2. \quad (22)$$

Segundo [Amadin e Obi 2015], o modelo ANFIS consiste de uma rede neural de duas fases, para frente (fase de ida) (propagation) e para trás (fase de retorno) (Back), portanto utiliza-se do modelo Backpropagation de Rede Neural para a aprendizagem da classificação de informações. A rede tem cinco camadas, como segue:

- Camada 1: O processo de nebulização transforma os valores nebulosos em termos linguísticos da função de relacionamento gaussiana mostrada pela Equação 23. Em que $\{a_i, b_i, c_i\}$ formam o conjunto de parâmetros, e o conjunto dos valores de entrada formado por: x ou y . Como os valores dos parâmetros mudam, a função sino formada varia conforme vão apresentando-se as várias formas das funções relacionamento para o conjunto nebuloso.

$$\mu_A(x) = \frac{1}{1 + \left(\frac{x-c_i}{a_i}\right)^{2b_i}} \quad (23)$$

- Camada 2: A fase de inferência utilizando o operador t-norma. Cada neurônio de saída representa o peso de ativação de uma regra. Em geral, qualquer outro operador \top – norma que executa o AND nebuloso pode ser usada como função do neurônio nesta camada.
- Camada 3: Calculando a relação dos pontos fortes das regras, através da normalização dos pesos utilizado a Equação 24. Em que $O_{3,i}$ é o um neurônio i da camada 3 que teve seu peso \bar{w}_i normalizado através dos pesos w_1 e w_2 .

$$O_{3,i} = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2. \quad (24)$$

- Camada 4: Calculando os parâmetros consequentes da regras nebulosas. Todos os neurônios i nesta camada são adaptativos com sua função calculada pela Equação 25. Onde \bar{w}_i é um peso normalizado da camada 3 e $\{p_i, q_i, r_i\}$ é o conjunto de parâmetros deste neurônio. Os parâmetros nesta camada são os consequentes de regra nebulosa.

$$O_{4,i} = \bar{w}_i f_i = \bar{w}_i (p_i x + q_i y + r_i), \quad (25)$$

- Camada 5: Cálculo da produção total como a soma de todos os sinais recebidos através da Equação 26.

$$\text{saída geral} = O_{5,1} = \sum_i \bar{w}_i \cdot f_i = \frac{\sum_i w_i \cdot f_i}{w_i} \quad (26)$$

A Figura 6(a) sintetiza o raciocínio abordado, para a arquitetura que corresponde ao sistema ANFIS, Figura 6(b).

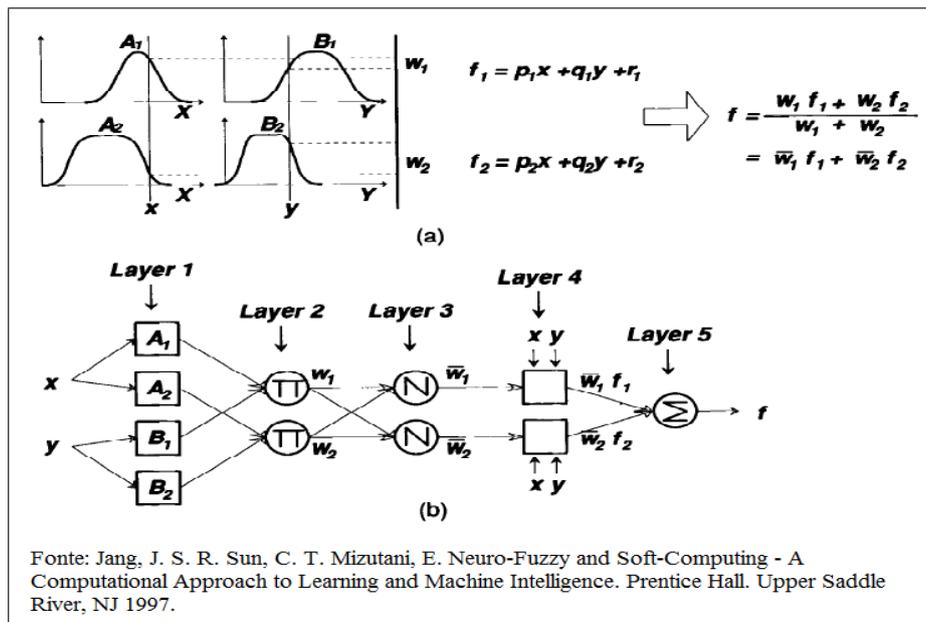


Figura 6. Modelo ANFIS (a) Um modelo TSK Nebuloso de primeira ordem com duas regras (b) A arquitetura ANFIS correspondente.

6. Modelo Nebuloso de Aluno

A abordagem utilizada para classificação do desempenho dos alunos leva em consideração atributos numéricos como: nota, assiduidade (pontualidade) e número de reprovações. Tais atributos formarão a base do raciocínio nebuloso adotado para classificação/qualificação do desempenho de alunos. O raciocínio aproximado, isto é, utilizado a lógica nebulosa, permite a criação de sistemas que classificação/qualificação informações e as aprendem baseado na maneira como professores (humanos) pensam, quando necessitam qualificar a aprendizagem de seus alunos. Os professores

costumam conceituar o nível de aprendizagem de seus alunos como: ótimo, bom, regular, ruim, péssimo, entre outros. O classificador neuro-nebuloso desenvolvido nesta pesquisa utiliza os atributos (nota, assiduidade e número de reprovações) para conceituar o desempenho dos alunos (Ótimo, Bom, Regular, Ruim e Péssimo).

De acordo com [Sevarac 2006] o objetivo da modelagem nebulosa do estudante é similar o modelo estratégico de aluno utilizado pelo professor (humano). Professores humanos não constroem modelos detalhados para compreensão do desempenho dos alunos para adaptar sua estratégia de ensino. Eles reúnem informações, e formam ideias gerais de que tipo de ensino pode funcionar melhor para cada aluno. De acordo com algumas descobertas os alunos são geralmente classificados em termos de algumas dimensões, como motivação subjacente, capacidade intelectual e nível de conhecimento sobre algum tema. Esse tipo de classificação pode ser facilmente expressa em termos de lógica nebulosa.

A classificação dos alunos pode ser desenvolvida baseada em regras nebulosas, como por exemplo:

SE Nota é Alta E Assiduidade é Alta THEN Aluno é ótimo

SE Nota é Média E Assiduidade é Alta E Reprovação é Baixa THEN Aluno é Bom

Segundo [Zadeh 1975] os conjuntos nebulosos (Nota, Assiduidade e Reprovação) são representados por variáveis linguísticas, pois expressam diretamente uma ideia de quantificação imprecisa (Alto, Médio, Baixo, etc). As variáveis linguísticas do modelo de aluno tem valores nebulosos (Alta, Média e Baixa) para classificar o conceito dos alunos. De acordo com as regras regras nebulosas um aluno que tem nota Alta e assiduidade Alta é considerado um ótimo aluno. Um aluno com nota Média, assiduidade Alta e Baixa reprovação é classificado como um Bom aluno.

Com base nestas regras de classificação o modelo ANFIS é implementado, porém as variáveis nebulosas precisam ser representadas por valores numéricos, por exemplo: será considerada nota Alta qualquer valor entre 8.0 e 10.0, em que 10.0 é o valor máximo e 8.0 nota mínima do conceito nota Alta. O modelo ANFIS implementado leva em consideração os valores contidos nas tabelas (Tabela 1), (Tabela 2) e (Tabela 3), para classificar os atributos nebulosos e um atributo Classe Conceito (Bom, Médio ou Ruim), respectivamente.

Nota	Máximo	Mínimo
Alta	10.0	8.0
Média	7.9	6.0
Baixa	5.9	0.0

Tabela 1. Valores da variável linguística Nota

7. Implementação de Um Classificador Neuro-nebuloso Utilizando o ANFIS

Nesta seção, é desenvolvido uma aplicação como o pacote FRBS (Fuzzy Rule-Based Systems), na linguagem R, para classificação de dados nebulosos. O dataset, utilizado foi desenvolvido, pelo autor, com base nas informações (atributos/valores de classificação) de 500 (quinhentos) alunos, sendo que 400 (quatrocentos) registros

Reprovação	Máximo	Mínimo
Alta	5	4
Média	3	2
Baixa	1	0

Tabela 2. Valores da variável linguística *Reprovação*

Assiduidade (%)	Máximo	Mínimo
Alta	100.0	90.0
Média	89.0	75.0
Baixa	74.0	0.0

Tabela 3. Valores da variável linguística *Assiduidade* (%)

foram utilizados para o treinamento do modelo implementado e 100 (cem) usados para o teste do modelo neuro-fuzzy de classificação. Os critérios para montagem do dataset levou em consideração que o desempenho escolar é medido através das notas e assiduidades de alunos de um curso/disciplina ministrada de maneira presencial.

As variáveis numéricas utilizadas na classificação serão utilizadas como exemplo de aprendizagem do modelo ANFIS. O atributo classe nebuloso Conceito contém os valores que serão “ensinados” ao classificador. Os atributos de classificação tem valores numéricos como entrada do sistema e produz como saída, para cada registro, um valor de classificação nebuloso para cada atributo classe. O dataset é o conjunto de dados (registros) organizados por linhas e colunas contendo os dados de uma determinada quantidade de alunos de uma escola. Cada atributo de classificação (nota, assiduidade e o número de reprovações) e o atributo classe (Conceito) são os exemplos de aprendizagem do modelo, por tanto, o classificador ANFIS implementado é um modelo de aprendizagem supervisionada de máquina.

Existem quatro passos para simular a obtenção dos resultados. Em primeiro lugar, a etapa de pré-processamento que prepara os dados como entrada do método. Em seguida, é gerado o modelo ou objeto. Em terceiro lugar, é feita uma previsão dos dados utilizando novos dados. Finalmente, a impressão dos resultados e o cálculo do erro entre o valor real e os valores previstos. Além disso, é descrito como usar o pacote FRBS da linguagem R, passo a passo, a fim de ajudar os usuários que não são familiarizados com o pacote de classificação da linguagem R.

Na etapa de pré-processamento, os dados são divididos em duas partes: dados de treinamento e de teste. Os dados de treinamento são usados para gerar sistemas baseados em regras nebulosas. O pacote FRBS, através das funções de normalização e raciocínio nebuloso, converte os valores nebulosos em dados numéricos. Por exemplo, neste caso, esta sendo utilizado um conjunto de dados com os valores dos atributos de classificação numéricos e valores nebulosos do atributo classe contidos no DatasetAlunos.data que tem valores: “Ótimo”, “Bom”, “Regular”, “Ruim” e “Péssimo”, estes “rótulos” são convertidos em valores numéricos inteiros, por exemplo: 1-péssimo, 2-ruim, 3-regular, 4-bom e 5-otimo. Além disso, os valores de saída que serão previstos devem ser colocados na última coluna do dataset.

A faixa de dados pode ser definida ou omitida. Se for omitida, o FRBS irá



calcular o mínimo e máximo de dados de treinamento do intervalo, é fortemente recomendado que este parâmetro deva ser definido para evitar que alguns dados fora da faixa, especialmente quando é realizada a fase de previsão dos dados, seja computados como reais. Devem-se levar em conta que, para o caso de tarefas de classificação, o intervalo de dados é apenas para as variáveis de entrada. Esta fase é implementada pelo script 7.1 em R:

Listing 1. Script 7.1

```
1 install.packages('frbs')
2 library("frbs")
3 alunos=read.table(file="F://...//DatasetAlunos.data")
4 alunosShuffled<-alunos[sample(nrow(alunos)),]
5 alunosShuffled[,4]<-unclass(alunosShuffled[,4])
6 tra.alunos<-alunosShuffled[1:400,]
7 tst.alunos<-alunosShuffled[401:nrow(alunosShuffled),1:3]
8 real.alunos<-matrix(alunosShuffled[401:nrow(alunosShuffled),4],ncol=1)
```

Observando o script 7.1, na primeira linha é instalado o pacote FRBS que contém as funções que implementam o modelo ANFIS. Na segunda linha do script 7.1 é definida a biblioteca FRBS responsável por criar o modelo TSK de classificação. Na terceira linha, do script 7.1, estão os dados contidos no arquivo *DatasetAlunos.data* localizado no diretório *F://.../DatasetAlunos.data* e salva-os no objeto *alunos*, ou seja, a função *read.table* carrega o arquivo que contém as informações do dataset e salva-os na variável *alunos* que serão o alvo do modelos de classificação. A quarta linha, do script 7.1, é responsável por separar os dados dos atributos de classificação dos atributos classe nebulosos e salva-los no objeto *alunosShuffled*. A quinta linha do script 7.1 informa quantas e quais colunas o arquivo tem e quais colunas são atributos de classificação e quais são os atributos classes. Na linha seis do script 7.1 informa-se quais serão os dados de treinamento e na linha sete do script 7.1 quais dados do Dataset serão os dados de validação do modelo e por fim na linha oito informa-se que a coluna quatro contém os rótulos de classificação de cada registro.

Em seguida, na etapa de geração do modelo (script 7.2), necessitam-se definir alguns parâmetros, que são: determinar o intervalo de valores numéricos máximo e mínimo de cada atributo, isso é realizado preenchendo a variável *matrix* com os valores de máximo e mínimo contidos no vetor $c(0, 10, 0, 100, 0, 14)$ de duas linhas ($nrow = 2$). Na segunda linha do, script 7.2, método *method.type*: é usado para determinar o método de aprendizagem. Neste exemplo é executado o método (Fuzzy Rule-Based Classification System - FRCS), com base na técnica de Ishibuchi e Nakashima (2001) (para a geração de regras a partir dos pesos ponderados) *FRBCS.W*. Assim, este parâmetro é atribuído a *FRBCS.W*. Na linha 3, do script 7.2, o comando *control*: cria uma lista de todos os parâmetros nebulosos, dependendo do algoritmo de aprendizagem a ser usado. Uma lista completa de parâmetros pode ser vista no manual FRBS disponível no link da comunidade de desenvolvedores CRAN R: <http://cran.r-project.org/web/packages/frbs/frbs.pdf>. Nesta lista, estão definidos os seguintes parâmetros: *num.labels* - é um inteiro positivo para determinar o número de rótulos/termos nebulosos em que o valor padrão é 7, porém adotaremos o valor 3 pois este é o número de rótulos nebulosos utilizados Alta-

Média-Baixa), *type.mf* - são as formas das funções de pertinência. Neste caso, as formas utilizadas foram funções de Gauss, *type.tnorm* - é o tipo de operador de conjunção (t-norma) neste caso, é definido *MIN* para ser tipo de t-norma, *type.snorm* - é o tipo de operador de disjunção (s-norma), neste caso, é definido *MAX* para ser tipo de t-norma e *type.implication.func* - é o tipo de função implicação. Neste caso, definiu-se *ZADEH* para ser tipo de t-norma (script 7.2).

O modelo de aprendizagem artificial utiliza o pacote *frbs.lean* (quarta linha do script 7.2), esta é a principal função de classificação ANFIS, pois esta implementa a rede neural artificial que aprende a classificar os dados através dos exemplos apresentados (Aprendizagem Supervisionada). O pacote *frbs.lean* utiliza a lista de controle: *control*, o intervalo de valores máximos e mínimos *range.data.input*, o tipo do método de aprendizagem nebulosa *method.type*, a lista de parâmetros nebulosos de controle *control* e os dados de treinamento contidos da variável *tra.alunos* para configurar o modelo de rede neuro-fuzzy projetada. O script 7.2 implementa esta fase.

Listing 2. Script 7.2

```
1 range.data.input<-matrix(c(4.5, 10, 8, 16, 0, 14), nrow=2)
2 method.type<-"FRBCS.W"
3 control<-list(num.labels=3,type.mf="GAUSSIAN",type.tnorm="MIN",type.
  snorm="MAX",type.implication.func="ZADEH")
4 object.cls<-frbs.learn(tra.alunos, range.data.input, method.type,
  control)
```

Em seguida, o modelo gerado é treinado e usado para prever a classificação dos dados de teste. Os dados a serem testados estão contidos na variável *tst.alunos*. A previsão de classificação utiliza o modelo previamente treinado e armazenado na variável *object.cls* para prever a classificação dos dados de teste. A função *predict* aplica os teste com relação ao modelo criado de classificação neuro-fuzzy. O script 7.3 implementa a etapa de previsão de classificação dos dados.

Listing 3. Script 7.3

```
1 res.test <- predict(object.cls, tst.alunos)
```

Por fim é implementada a etapa de impressão dos resultados e plotagem dos gráficos gerados do modelo de classificação neuro-fuzzy, o script 7.4 implementa esta fase.

Listing 4. Script 7.4

```
1 summary(object.cls)
2 plotMF(object.cls)
3 #[script 7.4]
```

A função *summary(object.cls)* sumariza o modelo de aprendizagem com base nos dados do dataset, a função *plotMF(object.cls)* gera o gráfico da função relacionamento dos rótulos nebulosos do dataset *alunos.txt*, a Figura 7

O comando *res.test* no interpretador da linguagem R, imprime a classificação dos dados de teste e validação contidos no dataset como resultado final e se for digitado o comando *plot(res.test)* é plotado um gráfico com os resultados do conjunto

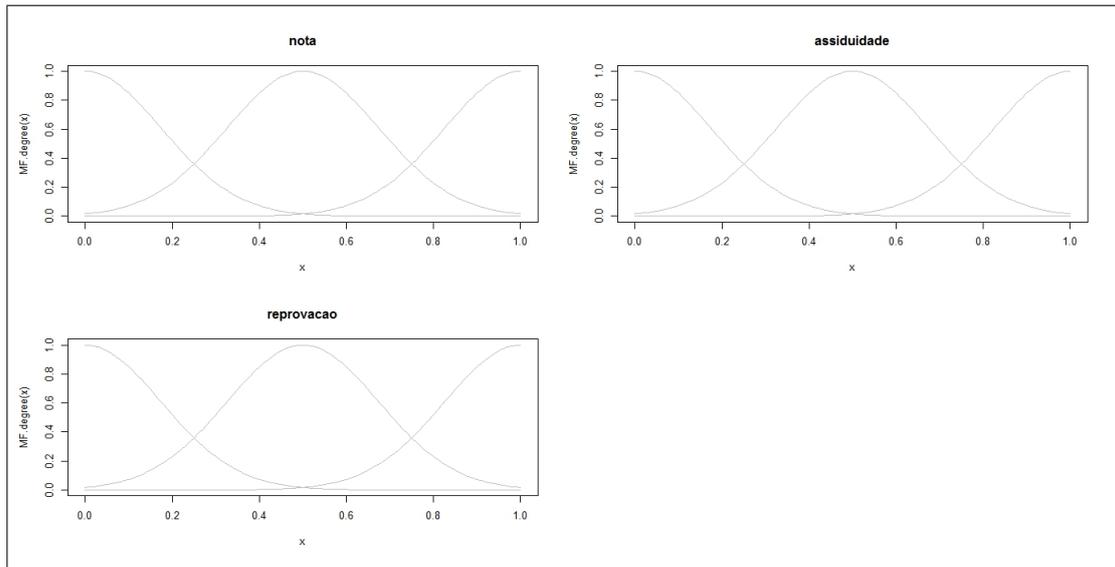


Figura 7. Gráfico da função relacionamento dos atributos/rótulos nebulosos.

classificado pelo modelo neuro-nebuloso implementado. A Figura 8 mostra o resultado dos dados classificados pelo modelo implementado nesta pesquisa.

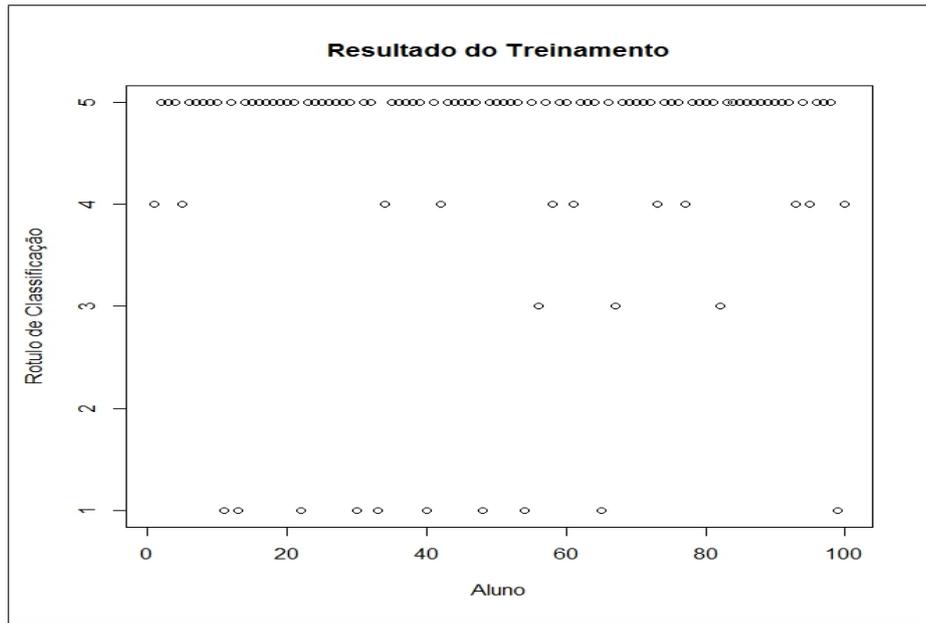


Figura 8. Gráfico dos dados classificados pelo modelo neuro-nebuloso.

8. Conclusão

As arquiteturas neuro-nebulosas utilizam os modelos de aprendizagem das Redes Neurais Artificiais para determinação de parâmetros dos Sistemas de Inferência Nebulosa (FIS-Fuzzy Inference System). O sistema neuro-nebuloso de classificação ANFIS implementa os métodos lineares de aprendizagem, ou seja, as regras nebulosas são compostas por polinômios lineares, para calcular o peso de cada valor de



entrada na rede, nas consequências das regras nebulosas. O modelo ANFIS implementado nesta pesquisa cumpre o papel de classificação de informações não-lineares uma vez que as informações.

A classificação de informações não lineares através de parâmetros linguisticamente nebulosos é possível devido a normalização dos dados na camada de entrada da rede. Após a normalização dos dados as camadas ocultas da rede ajustam os pesos sinápticos dos neurônios que compõe a rede. O classificador neuro-nebuloso da linguagem R possibilita o treinamento, o teste e a validação dos dados classificados através do pacote frbs.

Por se tratar de um modelo neuro-nebuloso que aprende a classificar com exemplos, foi observado que 53 alunos tiveram ter seus conceitos classificados de forma imprópria pelo modelo implementado. Fato este que não, necessariamente, implica em falha e sim em um conjunto de exemplos de aprendizagem (dataset) pequeno e composto de regras nebulosas insuficientes para classificação adequada. [Abraham e Obi 2015] afirmam que faltam estudos para definição de métricas de avaliação do desempenho dos frameworks FIS. O fato de o modelo ANFIS implementar uma de rede neural tipo Backpropagation com heurística nebulosa, ou seja, com aprendizagem supervisionada, torna sua eficiência de classificação fortemente relacionada a quantidade de exemplos nebulosos que o modelo é submetido, quanto mais exemplos o sistema é submetido, mais se aprende e mais adequadamente é a classificação conceitual dos alunos.

Referências

- [1] Abraham, A. Neuro fuzzy systems: State-of-the-art modeling techniques. CoRR, Churchill-AU, cs.AI/0405011, <http://arxiv.org/abs/cs.AI/0405011>, 2004.
- [2] Almeida, P. E. M. Evsukoff, A. G. Sistemas Fuzzy. In: Rezende, S. O. Sistemas Inteligentes: fundamentos e aplicações. Barueri, SP: Manole, Cap.7, p. 169-201, 2003.
- [3] Amadin, F. I. Obi, J. C. An Objective Approach to Schizophrenia Recognition Utilizing an Adaptive Neuro-Fuzzy Inference (ANFIS) Mode. Transactions on Machine Learning And Artificial Intelligence. Benin-Nigéria, Vol 3, N° 2, pp 01, 2015.
- [4] Amadin, F. I. Obi, J. C. English Premier League (EPL) Soccer Matches Prediction using An Adaptive Neuro-Fuzzy Inference System (ANFIS) for. Benin-Nigéria, Vol 3, N° 2, pp 24, 2015.
- [5] Barbosa, A. H, Freitas, M. S. R, Neves, F. A. Confiabilidade estrutural utilizando o método de Monte Carlo e Redes Neurais. REM, 58 3. Ouro Preto-Brasil, 2005.
- [6] Coppin, B. Inteligência Artificial. LTC, 2012.
- [7] Faceli, K. Lorena, A. C. Gama, J. Carvalho, A. C. P. L. F. Inteligência Artificial – Uma Abordagem de Aprendizado de Máquina. 1. ed. Rio de Janeiro: LTC, 2011. v. 1. 394 p.
- [8] Gomide, F. Gudwin R. Tanscheit, R. Conceitos fundamentais da teoria de conjuntos fuzzy, Lógica Fuzzy e Aplicações. In: 6th IFSA Congress-Tutorials, 1-38, 1995.



- [9] Haykin, S. *Redes Neurais: Princípios e Prática*. 2^a ed, Porto Alegre: Bookman, 2001.p 183 – 278.
- [10] Ishibuchi, H. Nakashima, T. Effect of rule weights in fuzzy rule-based classification systems. *IEEE Trans. on Fuzzy Systems*, v. 9, p. 506-515, 2001.
- [11] Jang, J. S. R. Sun, C. T. Mizutani, E. *Neuro-Fuzzy and Soft-Computing - A Computational Approach to Learning and Machine Intelligence*. Prentice Hall. Upper Saddle River, NJ 1997.
- [12] Kaufmann, A. Gupta, M. M. *Introduction to Fuzzy Arithmetic: Theory and Applications*, 2nd ed., Van Nostrand-Reinhold, New York, 1991.
- [13] Lin, C. T, Lee, C. S. G. *Neural Fuzzy Systems: a neuro-fuzzy synergism to intelligent systems*. Prentice-Hall, 1996.
- [14] Matloff, N. *The Art of R Programming. A Tour of statistical software design*. No Starch Press. San Francisco. 2011.
- [15] Mamdani, E. H. Application of fuzzy logic to approximate reasoning. *IEEE Trans. Comput.* 26:1182-1191, 1977.
- [16] McCulloch, W. S. Pitts, W. A logical calculus of the ideas imminent in nervous activity. *Bull. Math. Biol.* 5:115-133, 1943.
- [17] Neto, L. G. P. Nicoletti, M. C. *Introdução às Redes Neurais Construtivas*. 1^a ed, EdUFScar: São Carlos-SP, 2005.
- [18] Ross, T. J. *Fuzzy Logic with Engineering Applications*. 3^a ed, John Wiley and Sons, 2010.
- [19] Sevarac, Z. Neuro Fuzzy Reasoner for Student Modeling. In *The 6th IEEE International Conference on Advanced Learning Technologies (ICALT 2006)*, Kerkrade, The Netherlands, 2006.
- [20] Takagi, H. Fusion technology of fuzzy theory and neural networks: survey and future directions. In *Proceedings of International Conference on Fuzzy Logic and Neural Networks*, Lizuka, Japan, 1990, Vol. 1, pp. 13-26.
- [21] Tanaka, K. *An Introduction to fuzzy logic for practical applications*. N.Y.: Springer, 1997.
- [22] Widrow, B. Learning phenomena in layered neural networks. In *Proceedings of the IEEE First International Conference on Neural Networks*, 1987, pp. 411-430.
- [23] Zadeh, L. A. Fuzzy Sets. *Information and Control*. 8, 338-353, 1965.
- [24] Zadeh, L. A. Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Trans. Syst., Man, Cybernet.* 3(1):28-44, 1973.
- [25] Zadeh, L. A. The concept of a linguistic variable and its applications to approximate reasoning I, II, III. *Inform. Sci.* 8:199-249, 301-357; 9, 43-80, 1975.