



## EXTRAÇÃO DE REGRAS DE REDES SEM PESO WISARD ATRAVÉS DE SUAS IMAGENS “MENTAIS”

Paulo Vítor de Souza Coutinho

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia de Sistemas e Computação.

Orientador: Felipe Maia Galvão França

Rio de Janeiro  
Setembro de 2015

EXTRAÇÃO DE REGRAS DE REDES SEM PESO WISARD ATRAVÉS DE  
SUAS IMAGENS “MENTAIS”

Paulo Vítor de Souza Coutinho

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO  
ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE  
ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE  
JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A  
OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE  
SISTEMAS E COMPUTAÇÃO.

Examinada por:

---

Prof. Felipe Maia Galvão França, Ph.D.

---

Prof. Geraldo Bonorino Xexéo, D.Sc.

---

Prof. Josefino Cabral Melo Lima, Docteur

RIO DE JANEIRO, RJ – BRASIL  
SETEMBRO DE 2015

Coutinho, Paulo Vítor de Souza

Extração de Regras de Redes Sem Peso WiSARD através de suas Imagens “Mentais” / Paulo Vítor de Souza Coutinho. – Rio de Janeiro: UFRJ/COPPE, 2015.

XI, 34 p.: il.; 29, 7cm.

Orientador: Felipe Maia Galvão França

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia de Sistemas e Computação, 2015.

Referências Bibliográficas: p. 31 – 34.

1. WiSARD. 2. escala termômetro. 3. extração de regras. I. França, Felipe Maia Galvão. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.



# Agradecimentos

Dedico este trabalho primeiramente à minha família, meu pais Daise Silveira de Souza Coutinho e Luciano Guerra Coutinho, meu irmão Vitor Hugo de Souza Coutinho e aos demais familiares, por todo apoio e força dado nesta caminhada.

A todos os professores que tive em minha formação acadêmica, do ensino básico, do Oscar Tenório e da Uerj, incluindo Vera Werneck, Maria Alice Brito, Leandro Marzulo, Alexandre Sztajnberg, Paulo Eustáquio, Antônio Ritto, Guilherme Mota, Mariluci Portes (*in memoriam*), Roseli Wedemann, Rosa Costa, Neide Santos, Maria Clicia Castro, Ângelo Vimeney, Pedro Oscar entre outros, e a todo corpo docente da COPPE UFRJ, incluindo Carlos Eduardo Pedreira, Celina Miraglia Herrera de Figueiredo, Claudio Esperança, Geraldo Bonorino Xexéo, Gerson Zaverucha, Luis Felipe Magalhães de Moraes, Sulamita Klein, Luiz Pereira Calôba.

Ao meu orientador, Felipe Maia Galvão França, que me acompanhou nesta jornada e me deu todo apoio ao redigir esta dissertação.

A todos os colegas que encontrei nesta caminhada e que me ajudaram de alguma forma, incluindo Israel Zinc, Denilson Tavares, Jefferson Elbert, Victor Marconi, Michele Trindade, Evandro Macedo, Matheus Correia, Brunno Goldstein, Leonardo Nardi, Ingrid Boesing Paixão, Jean Pacheco, Ariela Damasceno, Ana Maria Silva, Bruno Brazil, Moysés Sampaio, Jaime Joaquim, Guilherme de Leo Silva, Roger Brock, Carolina Sacramento, Gabriela Cruz, Ingrid Keller, Aline Rodrigues, Jessica Monçores, Raquel Cupolillo, Bernardo Arume, Izabela Gomes, Guilherme Bechtinger, Alexandre Maia, Victor Peres, Wellington Costa, Santerre Batalha, Diogo Soares, Gabriele de Moraes.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

## EXTRAÇÃO DE REGRAS DE REDES SEM PESO WISARD ATRAVÉS DE SUAS IMAGENS “MENTAIS”

Paulo Vítor de Souza Coutinho

Setembro/2015

Orientador: Felipe Maia Galvão França

Programa: Engenharia de Sistemas e Computação

Nesta dissertação, baseada no artigo publicado no ESANN de 2014, é apresentada uma nova utilização de redes neurais WiSARD para extração de regras. Essas redes utilizariam, como entrada, representações especiais de dados, em escala termômetro, obtidas através de valores variáveis reais. Desenvolveu-se um algoritmo cuja tarefa era automatizar certas atividades que eram feitas manualmente no artigo-base. Dentre essas atividades estão a escolha de variáveis comparativas e das desigualdades para a confecção das regras. Vários testes, incluindo como parâmetros diferentes números de escalas e tipos de concatenação E ou OU, mostraram o alcance de bons resultados, ainda que pouco abaixo do estado da arte.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

RULE EXTRACTION FROM WISARD WEIGHTLESS NETWORKS  
THROUGH THEIR “MENTAL” IMAGES

Paulo Vítor de Souza Coutinho

September/2015

Advisor: Felipe Maia Galvão França

Department: Systems Engineering and Computer Science

In this work, based on a published article for ESANN 2014, it is presented a novel utilization of WiSARD neural networks for rule extraction. These networks would use, as entrance, special data representations, in thermometer scale, obtained through variable float values. An algorithm was developed whose task was to automate certain manual activities made on the base article. Among those activities are comparative variables and unequalities choice for rule confection. Several tests, including different number of scales and AND or OR concatenation types as parameters, showed the reach of good results, yet a bit below from state-of-art.

# Sumário

<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Objetivo . . . . .	2
1.2 Organização do trabalho . . . . .	2
<b>2 Redes Neurais Sem Peso e WiSARD</b>	<b>4</b>
2.1 As Redes Neurais Computacionais . . . . .	4
2.2 Redes Neurais Sem Peso . . . . .	4
2.3 WiSARD . . . . .	5
2.3.1 Histórico . . . . .	5
2.3.2 Conceito . . . . .	5
2.3.3 Imagens Mentais . . . . .	9
<b>3 Extração de Regras</b>	<b>11</b>
3.1 Conceitos Teóricos . . . . .	11
3.2 O estado da arte em extração de regras . . . . .	13
3.2.1 Engenharia reversa de redes neurais . . . . .	13
3.2.2 Algoritmos evolucionários . . . . .	14
<b>4 Método Proposto</b>	<b>17</b>
4.1 Bases de dados . . . . .	17
4.1.1 <i>Iris</i> . . . . .	17
4.1.2 Câncer de mama . . . . .	17
4.1.3 Vinhos . . . . .	18
4.2 Estruturando as redes WiSARD . . . . .	18
4.2.1 A escala termômetro . . . . .	19
4.3 Algoritmos propostos . . . . .	20
4.3.1 Construção das regras . . . . .	23



<b>5</b>	<b>Resultados e Discussão</b>	<b>25</b>
5.1	Metodologia . . . . .	25
5.1.1	Concatenações E e OU . . . . .	25
5.1.2	Rotina de testes . . . . .	26
5.2	Resultados experimentais . . . . .	26
5.2.1	Melhores resultados . . . . .	27
5.3	Discussão . . . . .	28
5.3.1	Comparação com o estado da arte dos algoritmos . . . . .	29
<b>6</b>	<b>Conclusão</b>	<b>30</b>
	<b>Referências Bibliográficas</b>	<b>31</b>

# Lista de Figuras

2.1	Estrutura de um perceptron . . . . .	4
2.2	Esquema de aprendizado de uma rede neural WiSARD . . . . .	6
2.3	Esquema de reconhecimento de padrão de uma rede neural WiSARD	7
2.4	Classificação de um padrão pela rede neural WiSARD . . . . .	7
2.5	Esquema de aprendizado de uma rede neural WiSARD com <i>bleaching</i>	8
2.6	Resultado de uma imagem pela rede neural WiSARD com <i>bleaching</i> .	9
2.7	Processo de formação da imagem mental a partir de um discriminador WiSARD treinado . . . . .	10
2.8	Estágios de aprendizado da rede neural WiSARD através da DRASiW	10
4.1	Padrão de entrada na escala termômetro para quatro variáveis . . . . .	19
4.2	Exemplo de uma imagem mental de uma rede treinada com escalas termômetro . . . . .	20
4.3	Comparação de valores entre imagens “mentais” . . . . .	21
5.1	Árvore de decisão das flores Iris . . . . .	27
5.2	Árvore de decisão do câncer de mama . . . . .	28
5.3	Árvore de decisão das vinícolas . . . . .	28

# Lista de Tabelas

3.1	Resultados do algoritmo RxREN . . . . .	13
3.2	Resultados do algoritmo de Wang . . . . .	14
3.3	Resultados do algoritmo G-REX . . . . .	15
3.4	Resultados do algoritmo CAREX . . . . .	16
3.5	Resultados do algoritmo de detecção de anomalias de Hou e Dozier . . . . .	16
5.1	Taxa de acerto do algoritmo proposto . . . . .	26
5.2	Comparação da taxa de acerto do algoritmo com a WiSARD . . . . .	27
5.3	Taxa de acerto dos algoritmos . . . . .	29

# Capítulo 1

## Introdução

Muito tem se pesquisado sobre a real importância e utilidade da extração de regras no contexto da inteligência artificial. Basicamente, isto tem servido como forma de compreender o aprendizado das redes neurais computacionais, principalmente as tradicionais.

Um assunto bastante estudado é como a informação é armazenada em redes neurais *feedforward*. A tarefa de se associar conexões a padrões assimilados pela rede a fim de se compreender o significado do aprendizado não tem sido das mais fáceis [1] [2].

A grande deficiência das redes neurais artificiais se encontra na tentativa de se compreender como estas redes adquiriam conhecimento, e que seria bastante desejável, senão essencial, que isto fizesse parte das funcionalidades de uma rede neural artificial treinada [3].

Redes neurais treinadas possuem uma grande quantidade de conexões, ocasionando uma compreensão ruim do aprendizado, isto é, pobre transparência de conhecimento e de habilidade de explicação, tornando a extração de regras um assunto crítico nos últimos anos. Foi proposto, então, o dilema da fidelidade-precisão, em que se discute a confusão que se faz entre as regras extraídas serem precisas ou serem fiéis à resposta da rede neural, e que dificilmente se alcança esses dois resultados [4].

Tarefas de classificação de dados também não são fáceis, principalmente quando há nuances durante a distinção entre uma classe e outra. Por exemplo, considerando a base de dados *Iris*, uma flor de uma classe  $A$  que possui comprimento de pétala  $x$  pode ser classificada erroneamente como sendo da classe  $B$  simplesmente porque flores dessa classe normalmente possuem aproximadamente este comprimento de pétala.

Muitos dos trabalhos já desenvolvidos em torno deste tema mostraram resultados com excelentes taxas de acerto [5] [6] [7] [8] [9]. Porém, nenhum deles consegue ser perfeito, devido às zonas de confusão que podem existir entre uma classe e outra, não sendo estas, portanto, completamente distinguíveis.

A respeito da extração de regras utilizando redes neurais sem peso, pode-se

dizer que esta é uma área ainda pouco explorada da Inteligência Artificial. Alguns trabalhos podem ser citados, como os que tratam diretamente imagens “mentais” de WiSARD para extração de regras fuzzy [10] [1]. Também temos como exemplo a extração de regras diretamente de padrões treinados pela rede neural sem peso [11].

Este trabalho é uma evolução do artigo desenvolvido para o ESANN de 2014, que teve como objetivo verificar a viabilidade de se poder extrair regras a partir de imagens “mentais” obtidas por redes neurais sem peso WiSARD treinadas. Foi proposto um novo meio de se chegar a esse resultado, através de representação de dados em escala termômetro. Essa é uma representação matricial bidimensional, onde cada linha é um termômetro que representa uma variável diferente da amostra.

A partir das imagens “mentais” dessas representações de dados, o algoritmo compara os intervalos de valores entre classes por meio de desvios padrões. Escolhem-se os maiores valores, que representam as maiores discrepâncias entre elementos de uma classe e das demais, como determinantes para a composição de regras.

## 1.1 Objetivo

Este trabalho tem como objetivos:

- Desenvolver um algoritmo que seja capaz de extrair regras de decisão a partir de imagens “mentais” em escala termômetro de uma rede neural WiSARD treinada e que reproduza os resultados do trabalho-base (escolha das variáveis e dos valores de corte);
- Obter, a partir das regras extraídas, uma taxa de acerto próxima do estado da arte para todas as bases de dados.

Além da base de dados presente no trabalho anterior (flores do gênero *Iris*), foram incluídas mais duas bases de dados (câncer de mama e vinhos) para se avaliar o modo como o algoritmo extrai essas regras e, ainda, se verificar qual a taxa de acerto que essas regras conseguiriam e comparar com o estado da arte.

## 1.2 Organização do trabalho

O trabalho é dividido em seis capítulos. Recomenda-se uma leitura sequencial para um melhor entendimento do mesmo.

No capítulo 2, encontra-se um resumo das redes neurais computacionais e das redes neurais sem peso, em especial a tecnologia que está por trás da rede neural WiSARD.

No capítulo 3, são apresentados diversos conceitos envolvendo extração de regras, incluindo questões, problemas, e características, bem como alguns trabalhos de classificação que servirão de comparação para o resultado com o que este trabalho propõe.

No capítulo 4, são mostrados os passos de treinamento da rede WiSARD, as bases de dados utilizadas para treinamento e o algoritmo de extração de regras.

No capítulo 5, é apresentada a metodologia da elaboração das regras. São também mostrados os testes realizados com as regras extraídas e as discussões em torno dos resultados encontrados.

O capítulo 6 se destina à conclusão deste trabalho, confrontando o material teórico com os resultados práticos e apresentando sugestões quanto a trabalhos futuros.

# Capítulo 2

## Redes Neurais Sem Peso e WiSARD

### 2.1 As Redes Neurais Computacionais

Redes neurais são estruturas que procuram simular o comportamento do cérebro humano, no que diz respeito a atividades como aquisição de conhecimento e processamento de informações, através de algoritmos computacionais [12]. Sua arquitetura é composta de unidades chamadas de perceptrons, que são estruturas similares ao neurônio biológico. As entradas do perceptron correspondem aos dendritos, que possuem pesos ajustáveis via treinamento, e as saídas, aos axônios. Além disso, o processamento pode ser feito aplicando-se uma função de excitação à soma das entradas do perceptron e do viés, que gera um valor à saída do neurônio [13].

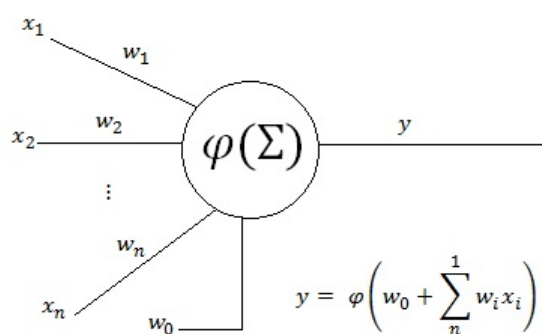


Figura 2.1: Estrutura de um perceptron

### 2.2 Redes Neurais Sem Peso

O conceito de rede neural sem peso surgiu baseado no modelo de arquitetura das memórias de acesso aleatório, conhecidas como RAM (Random Access Memory)

[14]. Elas se diferem das redes neurais tradicionais (perceptrons) pela ausência de pesos sinápticos nos dendritos, que são compensadas por essas estruturas RAM.

Desde então, diferentes modelos de redes neurais sem peso foram propostos, como o VG-RAM [15], SDM [16], GNU [17], GSN [14], PLN [18], pRAM [14], AutoWiSARD [19].

## 2.3 WiSARD

Redes neurais WiSARD são redes neurais sem peso que operam sob um vetor binário de  $X \times n$  elementos. A rede é composta por uma ou mais unidades básicas chamadas de discriminadores. Cada discriminador contém  $X$  RAMs que operam sobre  $n$  posições aleatórias do vetor, através de um mapeamento pseudo-aleatório.

### 2.3.1 Histórico

Três cientistas, Wilkie, Stonham e Alexander, propuseram um modelo de rede neural sem peso e construíram um protótipo na Universidade Brunel, sob licença da British Technology Group, em 1981. A arquitetura foi designada, a princípio, para o reconhecimento de imagens, inspeção e vigilância. Mas, com o passar dos anos, passou a ser usada com outros propósitos, entre eles reconhecimento linguístico.

Com relação ao modelo de neurônio artificial, formulado em 1943 pelos cientistas Warren S. McCulloch e Walter Pitts, este mostra ser mais eficaz, principalmente para o problema de separação tipo OU-exclusivo presente no modelo do perceptron, formulado por Papert e Minsky [13]. Enquanto que o neurônio tradicional, que utiliza sinais excitatórios e inibitórios, só é excitado se um sinal excitatório estiver ativo e um inibitório estiver inativo, o neurônio da WiSARD pode aceitar qualquer combinação de entradas para poder ser ativado [14].

### 2.3.2 Conceito

Uma RAM possui uma estrutura formada por um vetor binário de  $2^n$  posições, sendo  $n$  a quantidade de *bits* sobre os quais este atua. Cada posição da RAM representa uma combinação de valores que a tupla de *bits* mapeada possui.

O aprendizado ocorre quando padrões de imagem são processados pela rede. Inicialmente, o valor de todos os discriminadores é zerado, assim como os *bits* internos destes. Quando um novo padrão é aprendido, a combinação de valores de cada tupla estimula a posição correspondente de cada RAM, passando de '0' para '1'. O mesmo é feito para os demais discriminadores. A etapa termina quando os discriminadores da rede processam todos os padrões de imagem.



O reconhecimento de um padrão se dá comparando as tuplas com as posições encontradas em cada RAM. Estas RAMs retornam o valor treinado, que pode ser '0' ou '1', correspondente à tupla de valores encontrados no subconjunto mapeado. Os valores retornados pelas RAMs são somados, retornando-se um placar de reconhecimento da imagem pelo discriminador. Quanto maior for este placar, mais próxima esta imagem está do padrão treinado pelo discriminador.

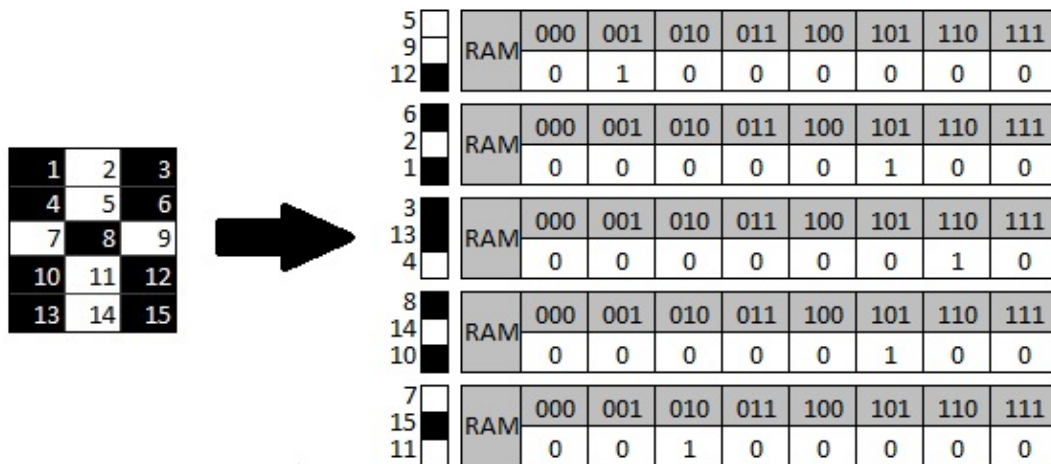


Figura 2.2: Esquema de aprendizado de uma rede neural WiSARD

A figura 2.2 mostra como um discriminador assimila uma imagem para a rede. Os números indicados de 1 a 12 representam o mapeamento pseudo-aleatório da rede. Cada RAM recebe valores de três posições e traduz para um número binário, variando de '000' até '111', totalizando 8 posições. A posição correspondente é alterada para '1', independente do valor anterior. O mesmo se repete para as outras RAMs.



placar mais alto dentre todos os demais. A rede, então, deduz a partir deste placar que a imagem pertence à classe de letras X.

Uma das vantagens de se usar redes WiSARD no reconhecimento de imagens é a sua tolerância a pequenos ruídos, principalmente quando as imagens treinadas não apresentam algum ruído. Entretanto, uma maior quantidade de ruído na imagem a ser reconhecida pode diminuir a confiança do resultado, comprometendo a sua classificação. Isto ocorre quando muitas posições de memória das RAMs estão assinaladas com '1', aumentando-se a chance de ocorrerem empates na classificação. A isto dá-se o nome de saturação. Como solução, propõe-se uma versão aperfeiçoada da WiSARD com a técnica de *bleaching*, que será explicada na próxima seção.

As redes WiSARD podem ser configuradas de diferentes maneiras. Elas podem ter poucas RAMs mapeando muitas posições de memória ou muitas RAMs com poucas posições mapeadas. As primeiras tendem a ser mais especificistas, mais intolerantes a ruídos, enquanto que as outras tendem a ser mais generalistas. Essa flexibilidade possibilitou diferentes usos dessa rede.

### *Bleaching*

O *bleaching* é uma técnica usada em redes WiSARD para desempatar discriminadores, gerando resultados mais decisivos. Consiste de uma variável adicional, chamada *b*, que interfere no resultado do discriminador. Essa variável determina a intensidade de *bleaching* no discriminador.

Com o *bleaching*, as RAMs teriam suas posições de memória não mais do tipo booleano ('0' ou '1'), mas do tipo inteiro positivo, contando quantas vezes aquela combinação de valores aparece na lista de padrões treinados.

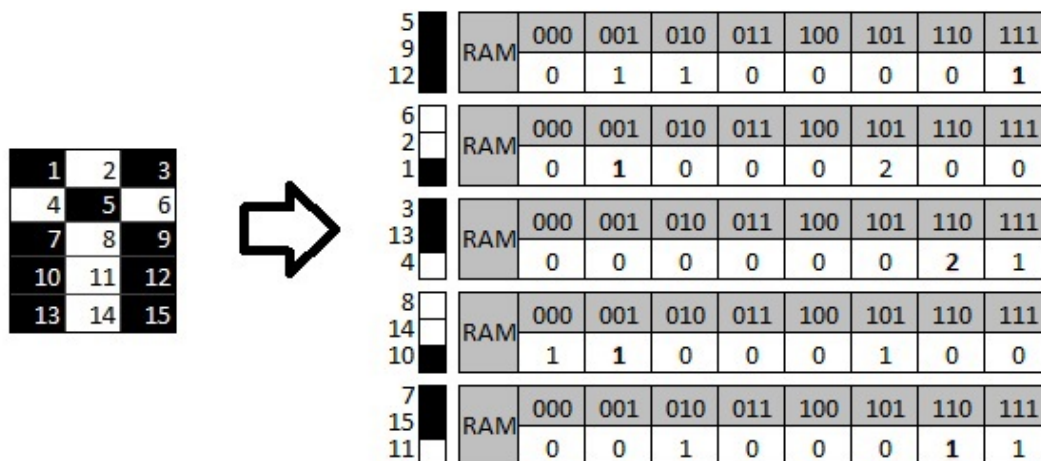


Figura 2.5: Esquema de aprendizado de uma rede neural WiSARD com *bleaching*

Na figura 2.5, vemos um exemplo de rede neural WiSARD modificada com *bleaching* recebendo um novo padrão. Os valores em negrito correspondem aos valores recém-atualizados. Note que combinações já contadas acabam tendo seus contadores incrementados, ao invés de permanecerem com o valor '1'.

Os resultados de cada RAM são transformados em valores binários dependendo do valor de *bleaching* submetido ao discriminador. Se o valor da RAM for menor que o nível de *bleaching*, o valor será 0; caso contrário, ele será 1. Todos os valores transformados são então somados, dando-se um novo resultado ao discriminador.

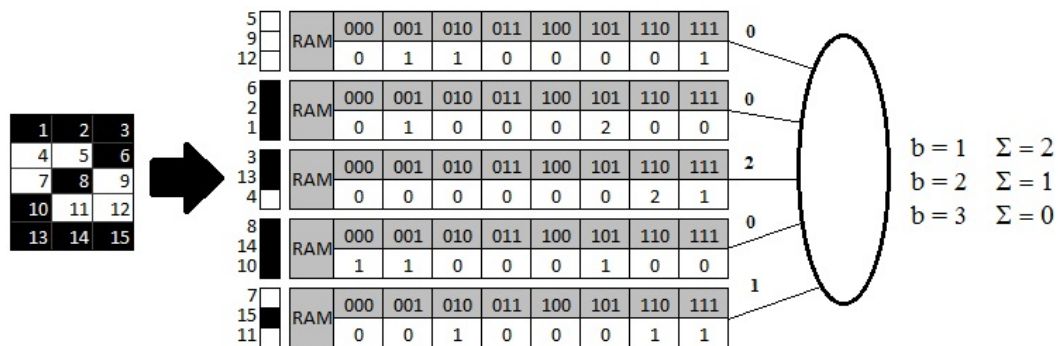


Figura 2.6: Resultado de uma imagem pela rede neural WiSARD com *bleaching*

Nessa figura, mostramos como uma rede WiSARD com *bleaching* faz o placar de uma imagem. Para valores mais altos, o placar se reduz, chegando a zero no caso de  $b = 3$ .

Caso ocorra empate entre os discriminadores, o valor da variável  $b$  é incrementado e os cálculos são refeitos. Segue-se até que os empates sejam desfeitos ou que o nível de *bleaching* atinja o máximo, fazendo com que a classificação seja inconclusiva.

### 2.3.3 Imagens Mentais

Imagens mentais são matrizes multidimensionais em escalas de cinza (valores reais entre 0 e 1) que representam o que a rede neural WiSARD aprendeu. São resultado de uma rede inversa chamada DRASiW. Todo esse processo ocorre graças ao mapeamento biunívoco, fornecendo como saída o aprendizado efetivo. Para que isto possa acontecer, é necessário que as RAMs tenham contadores e que se saiba o mapeamento da rede. Este processo, entretanto, não recupera cada imagem treinada pela rede, apenas reproduz a sobreposição das mesmas. Na imagem abaixo, é apresentado com detalhes como se obtém uma imagem mental da WiSARD.

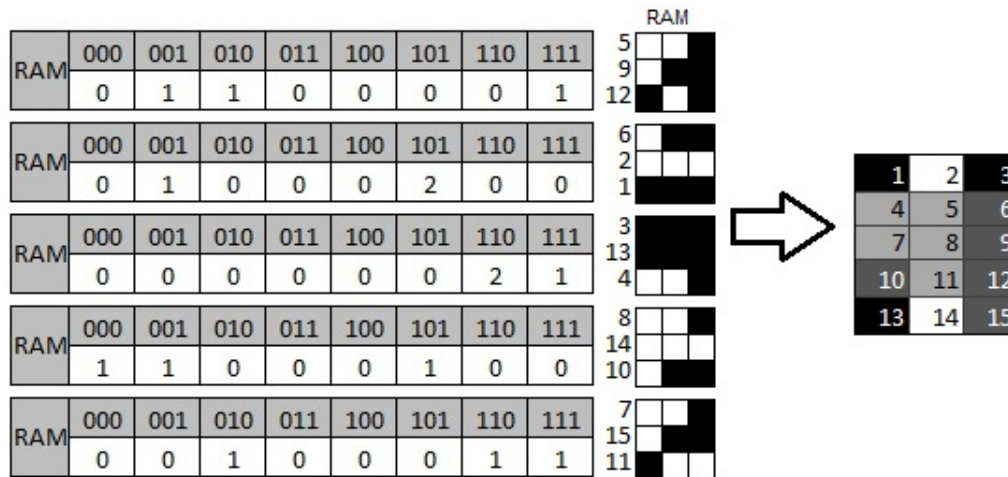


Figura 2.7: Processo de formação da imagem mental a partir de um discriminador WiSARD treinado

Pelo esquema apresentado na figura 2.7, três imagens foram processadas, visto o somatório de cada contador da RAM. Os números representam as posições mapeadas pela rede WiSARD. Dada uma matriz de valores inteiros com as mesmas dimensões de uma imagem treinada, e tendo seus valores zerados, os valores das RAMs são rearranjados pelas posições da imagem mental. A cada espaço hachurado, é incrementado em uma unidade a posição correspondente na matriz. Em seguida, tomando-se o número total de imagens aprendidas, esses elementos são divididos por esse valor, sendo transformados em valores racionais entre 0 e 1, que correspondem a tons em escala de cinza. Assim, a imagem gerada nesse processo é a imagem mental da rede nesse estado de treinamento.

A figura 2.8 mostra como a imagem mental da rede se modifica quando vários padrões são ensinados à rede. Neste exemplo, nota-se que para diferentes padrões de letras X, os cantos são mais comuns que o centro.

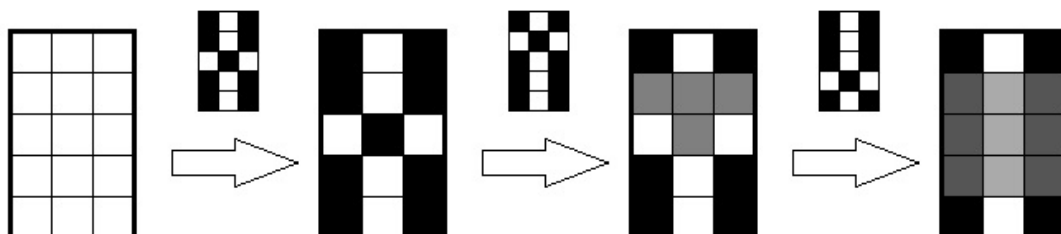


Figura 2.8: Estágios de aprendizado da rede neural WiSARD através da DRASiW

# Capítulo 3

## Extração de Regras

A extração de regras é uma das áreas mais estudadas no campo da Inteligência Artificial. Ela busca inferir características resumidas de um conjunto grande de dados, das quais não se conhecem essas características.

Vários recursos foram utilizados como meios de se obter regras, entre eles, algoritmos genéticos [8][20] [21] , redes neurais *feedforward* [3], mineração de dados [8] [7], algoritmos bacteriais para regras *fuzzy* [22], máquinas de vetores de suporte [23] [24] [25] etc.

Como exemplos de aplicabilidade de extração de regras, temos as áreas de finanças [26], linguística [27], medicina [28], da própria inteligência artificial [4] entre outros.

### 3.1 Conceitos Teóricos

A grande motivação de estudo dessa área surgiu pela natureza obscura das redes neurais tradicionais. Diversos trabalhos procuraram compreender por meio da extração de regras como redes neurais *feedforward* assimilavam a informação apresentada[3].

Zhu [4] mostra em seu artigo que há uma confusão entre se usar redes neurais *feedforward* como ferramenta para construção de regras ou como objeto para compreensão através das regras extraídas, dando-se o nome de dilema da fidelidade-accurácia.

Craven e Shavlik [29] definiram cinco critérios para a construção de bons algoritmos de extração de regras:

- **Compreensibilidade:** se tais representações são compreensíveis para os humanos;
- **Fidelidade:** se as representações são fiéis às redes pelas quais foram extraídas;
- **Precisão:** se tais representações fazem previsões para casos não vistos;

- **Escalabilidade:** se o método é capaz de se adaptar a redes com mais espaços de entrada e maior número de unidades e conexões com pesos;
- **Generalização:** se o método requer regimes de treinamento especial ou restrições na arquitetura da rede;

Desses critérios, os três primeiros têm sido os mais usados para pesquisas, enquanto que os dois últimos têm tido menos foco. Além disso, Craven e Shavlik também consideram um outro critério, disponibilidade de *software*, para o sucesso de um método de extração de regras. Este critério descreve que um algoritmo tenha seu código-fonte disponível, bem como este possa ser instalado por qualquer usuário em sua máquina.

Já Tickle e outros autores [30] define quatro critérios para se obter bons algoritmos de extração de regras:

- **Precisão:** diz respeito à capacidade do conjunto de regras de classificar corretamente um conjunto de exemplos não vistos, dentro do domínio do problema;
- **Fidelidade:** diz se as regras extraídas reproduzem o comportamento da rede neural de onde foram extraídas;
- **Consistência:** diz se diferentes sessões de treinamento conduzem a regras que realizam a mesma classificação de exemplares não vistos;
- **Compreensibilidade:** é a medição do conjunto de regras quanto ao número de regras e o número de antecedentes para cada regra;

Andrews e outros autores [3] classificam os algoritmos de extração de regras em três categorias, de acordo com a técnica utilizada:

- **Decomposicionais:** são algoritmos que extraem regras a nível de unidade (camada oculta e de saída) da rede;
- **Pedagógicos:** são aqueles que tratam a rede neural como sendo uma caixa-preta, ou seja, se baseiam apenas nos pares de entrada-saída da rede para realizar a extração de regras;
- **Ecléticos:** são aqueles que combinam características das duas categorias anteriores;

Andrews e outros autores [3] citam ainda que o conhecimento adquirido pela rede neural através da extração de regras deve ser codificado como a arquitetura da rede neural em si, a função de ativação associada com as camadas escondida e de saída da rede e com o conjunto de parâmetros da rede, também chamados de pesos.

## 3.2 O estado da arte em extração de regras

Nesta seção, apresentaremos alguns algoritmos de extração de regras pertencentes ao estado da arte, e que servirão de comparação com o algoritmo proposto neste trabalho.

### 3.2.1 Engenharia reversa de redes neurais

#### RxREN

RxREN (Engenharia reversa de redes neurais para extração de regras) [5] é um algoritmo de engenharia reversa que trabalha sobre redes neurais *feedforward*. O processo utiliza poda de redes neurais treinadas e cria regras a partir da rede podada, transformando a caixa-preta das redes neurais em caixa-branca. Este é um exemplo de utilização de extração de regras como forma de compreender o treinamento das redes neurais *feedforward*.

O algoritmo possui duas fases. Na primeira, o algoritmo remove as entradas dos neurônios que são irrelevantes e depois encontra nos remanescentes os dados que predominam em cada um deles. Essa tarefa cria uma matriz bidimensional, chamada de matriz de amplitude de dados. A matriz contém para cada neurônio de entrada e cada classe os valores mínimo e máximo dos respectivos dados que são representados.

A segunda fase consiste de três etapas: construção, poda e melhoria. Na construção, as regras são escritas com base na matriz de amplitude de dados, formando comparações com os valores mínimo e máximo da matriz. A poda remove as regras que menos contribuem para a acurácia do resultado e a melhoria busca melhorar ainda mais a classificação das regras remanescentes.

As regras extraídas pelo algoritmo possuem o seguinte formato:

$$\text{Se } (I_1 \geq i_1min \wedge I_1 \leq i_1max) \vee \dots \vee (I_n \geq i_nmin \wedge I_n \leq i_nmax) \text{ então } Classe = C1 \quad (3.1)$$

Onde os  $I$ 's representam os valores mínimo e máximo dos intervalos e  $C$  a classe a ser representada.

Para o experimento do algoritmo, foram utilizadas seis bases de dados, dentre elas *Iris* e câncer de mama. Os resultados foram considerados bons.

Tabela 3.1: Resultados do algoritmo RxREN

Base de Dados	Taxa de Acerto
<i>Iris</i>	97,3%
Câncer de mama	96,4%



## Algoritmo de Extração de Regras de Wang

O algoritmo de extração de regras proposto por Wang et. al. [6] utiliza treinamento de rede neural e poda para geração de regras.

O processo se inicia com a transformação dos valores da base de dados em valores binários (0 e 1) através de um método de clusterização e da utilização da escala termômetro. A próxima etapa é a definição do número de camadas ocultas da rede, que é feito através de validação cruzada. O resultado que obteve os melhores resultados foi o de três neurônios. A rede neural escolhida é treinada utilizando-se como função de otimização a soma da função de erro mínimo quadrático com o algoritmo de Levenberg-Marquadt, um algoritmo que possui aprendizado mais rápido e maior precisão que outros algoritmos de *backpropagation*.

O algoritmo de poda da rede funciona da seguinte forma:

- Escolhe-se um valor bem pequeno  $\sigma$  como viés de poda e um valor  $b$  como incremento desse viés;
- Escolhe-se um peso inicial aleatoriamente de uma rede treinada;
- Para cada peso  $w_i$  da rede, se este for menor que  $\sigma$ , então retire-o;
- Retreine a rede e faça  $\sigma = \sigma + b$ . Volte ao passo 3;

A poda da rede é uma maneira de simplificá-la para tornar-se mais fácil a extração das regras. As conexões menos importantes caem para próximo de zero, enquanto que as mais importantes são valorizadas.

Existe uma certa liberdade na confecção das regras, que se baseiam em escala termômetro para obtenção dos valores. Várias regras podem ser necessárias para se diferenciar uma classe das outras, como foi o caso da base de dados do câncer de mama.

Os experimentos feitos com esse algoritmos utilizaram duas bases de dados: *Iris* e câncer de mama. Os resultados são mostrados na tabela a seguir:

Tabela 3.2: Resultados do algoritmo de Wang

Base de Dados	Taxa de Acerto
<i>Iris</i>	98%
Câncer de mama	98,9%

### 3.2.2 Algoritmos evolucionários

#### G-REX

G-REX [7] é uma ferramenta de extração de regras que utiliza algoritmos genéticos, mais especificamente programação genética. Essa ferramenta, disponível na forma de

pacote Weka, adiciona funcionalidades à ferramenta de mineração de dados padrão Weka.

O algoritmo pode trabalhar tanto com dados numéricos quanto com *strings*. Para facilitar ainda mais a avaliação dos experimentos, pode-se usar a validação cruzada de  $N$  folhas com estratificação opcional.

As funções de aptidão do G-REX possuem duas propriedades: recompensa para bons resultados de otimização e punição para cada nó incluído na árvore. Mesmo que essas punições sejam efetivas em reduzir o tamanho dos modelos, ainda pode haver regras que seja duplicatas ou que não façam sentido.

G-REX tem diferentes modelos de classificação na sua implementação, que são:

- **Árvores de decisão;**
- **Lista de decisão** - diferem das árvores de decisão pelo fato dos nós decidirem por apenas uma classe;
- **k-Vizinho mais próximo;**
- **Lógica *fuzzy*** - neste, os atributos contínuos são *fuzzyficados* através de uma técnica baseada no algoritmo k-Médias;

No caso da extração de regras, a ferramenta permite uma flexibilidade na composição das mesmas, com conjunções e disjunções.

A ferramenta foi testada utilizando-se 17 bases de dados [31], entre os quais *Iris*, câncer de mama, vinhos e diabetes. Os resultados alcançados se encontram na tabela abaixo:

Tabela 3.3: Resultados do algoritmo G-REX

Base de Dados	Taxa de Acerto
<i>Iris</i>	92,0%
Câncer de mama	96,2%
Vinhos	92,5%

## CAREX

CAREX (Algoritmo Coevolucionário para Extração de Regras) [8] é um algoritmo proposto por Pawel Myszowski que combina mineração de dados e programação genética.

Dada uma população inicial de regras, cada indivíduo recebe um valor de função de aptidão que diz a qualidade da solução. Em seguida, o algoritmo checa se a condição de parada é satisfeita. Se não é satisfeita, ele utiliza operadores genéticos (mutação e crossover) para gerar uma nova população de indivíduos.

As regras são formadas por atributos **SE... ENTÃO**, onde as condições estão representadas por intervalos de dados, concatenados por conectores E, e as saídas pela classe correspondente.

Para o experimento do algoritmo, foram utilizadas as famosas bases de dados do UCI data repository para comparação de algoritmos, entre eles *Iris*, e vinhos; As instâncias, atributos e classes foram mantidas. O algoritmo se saiu melhor na base de dados *Iris*. A tabela abaixo mostra as taxas de acerto verificadas pelo algoritmo:

Tabela 3.4: Resultados do algoritmo CAREX

Base de Dados	Taxa de Acerto
<i>Iris</i>	96,67%
Vinhos	93,93%

### Sistema de classificação baseado em anomalias

Hou e Dozier [9] propuseram um sistema de detecção de anomalias baseado no seguinte problema: utilizando-se de amostras "normais" para constuir a função, determinar se uma amostra futura será "normal" ou "anormal".

O algoritmo de detecção de anomalias utiliza seleção negativa. Primeiramente, um conjunto gerado aleatoriamente de detectores é apresentado como conjunto candidato. Os detectores que combinarem com quaisquer amostras normais são substituídos por outros detectores gerados aleatoriamente. O processo continua até que um número válido de detectores seja definido.

As regras, convertidas a partir do sistema de detecção de anomalias, também utilizam intervalos de variáveis para comparações, mas concatena com disjunções (conectivo OU), como no exemplo abaixo:

$$\text{Se } (I_1 \geq i_1min \wedge I_1 \leq i_1max) \vee \dots \vee (I_n \geq i_nmin \wedge I_n \leq i_nmax) \text{ então } Classe = C1 \quad (3.2)$$

Os testes feitos com este algoritmo utilizaram como base de dados *Iris* e câncer de mama. As taxas de acerto foram acima de 99%, como mostrado na tabela abaixo:

Tabela 3.5: Resultados do algoritmo de detecção de anomalias de Hou e Dozier

Base de Dados	Taxa de Acerto
<i>Iris</i>	99,46%
Câncer de mama	99%

# Capítulo 4

## Método Proposto

Neste capítulo, é apresentado um método de extração de regras com base em redes neurais sem peso WiSARD. A ideia se baseou no trabalho proposto para o ESANN 2014 [32].

### 4.1 Bases de dados

#### 4.1.1 *Iris*

A famosa base de dados das flores do gênero *Iris* foi coletada por Sir Ronald Fisher em 1936. Ela é composta de 150 amostras, divididas em 3 conjuntos de 50 amostras cada, representando 3 espécies de *Iris*: *Iris setosa*, *Iris versicolor* e *Iris virginica*. As características analisadas foram largura e comprimento da sépala e largura e comprimento da pétala.

#### 4.1.2 Câncer de mama

Outra conhecida base de dados, coletada por William H. Wolberg, do Hospital da Universidade de Wisconsin, EUA, contém 699 amostras de tumores da mama, classificados como sendo benignos (458 amostras) e malignos (241 amostras). As 10 características abaixo foram mensuradas:

- Espessura;
- Uniformidade do tamanho da célula;
- Uniformidade do formato da célula;
- Adesão marginal;
- Tamanho da célula epitelial isolada;

- Núcleos descobertos;
- Cromatina;
- Nucléolos normais;
- Mitose.

### 4.1.3 Vinhos

A base de dados de vinhos apresenta 178 amostras de vinhos provenientes de 3 vincolas, às quais foram medidas 13 características:

- Teor alcoólico;
- Ácido málico;
- Cinza;
- Alcalinidade da cinza;
- Magnésio;
- Total de fenóis;
- Flavonoides;
- Fenóis não flavonoides;
- Proantocionidinas;
- Intensidade da cor;
- Tonalidade;
- OD280/OD315 de vinhos diluídos;
- Prolina.

## 4.2 Estruturando as redes WiSARD

Neste trabalho, foram levadas em consideração as seguintes configurações de rede WiSARD: número de *bits* por característica e número de linhas de entrada da RAM. Segundo o artigo do ESANN que originou este trabalho [32], a configuração de 1 linha de entrada por endereço e a de 3 *bits* por característica foi a que apresentou melhor capacidade de diferenciação de classes.

### 4.2.1 A escala termômetro

As redes WiSARD trabalham normalmente com vetores de dados binários ('0' e '1'). Para que pudéssemos trabalhar com variáveis de valores reais, foi feita uma adaptação nesses vetores: a utilização da escala termômetro como representação desses valores.

A escala termômetro é uma representação matricial binária de dados. As linhas representam as variáveis, que por sua vez possuem escalas, ou graus, lembrando um termômetro.

Qualquer valor real pode ser representado em escala termômetro, cuja nota depende de se estipular os valores mínimo e máximo da variável, numa espécie de discretização desse valor. Qualquer valor abaixo do mínimo faria com que o termômetro ficasse “vazio” e qualquer valor acima do máximo deixaria o termômetro “cheio”.

Assim, a escala termômetro pode ser traduzida como uma tupla de vetores contendo duas características: número de escalas e valor da escala atual.

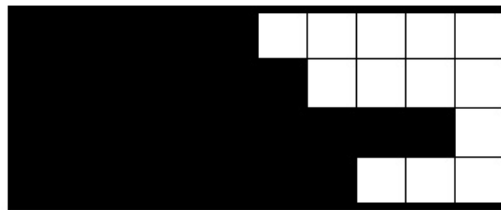


Figura 4.1: Padrão de entrada na escala termômetro para quatro variáveis

#### Transformação das variáveis

Para se adequar à escala termômetro, a base de dados foi tratada da seguinte forma:

- Para cada variável do conjunto inteiro de amostras, os valores máximo e mínimo são considerados.
- Dado um tamanho de escala, calcula-se para cada valor de variável o seu correspondente na escala atual através da seguinte fórmula:

$$escala_{atual} = \lfloor \frac{valor - minimo}{maximo - minimo} \rfloor \quad (4.1)$$

- Por último, as variáveis tratadas de cada amostra são empilhadas, formando um padrão bidimensional na escala termômetro.

Como exemplo, suponha o valor 0,6 numa escala que vai de 0 a 2,5. Esse valor, numa escala de 5, corresponderia ao nível 1 da escala.

Esse tratamento foi necessário para que não houvesse amostras que tivessem termômetros “cheios” ou “vazios”, interferindo na tarefa da extração de regras.

Nestres trabalho, foi adotada a distribuição uniforme de variáveis, ou seja, cada escala possui uma diferença igual entre os valores mínimo e máximo correspondentes ao tamanho da escala.

### Imagens mentais da escala termômetro

Um fato interessante que pode ser observado é que redes treinadas apenas com padrões que obedecem à escala termômetro, quando sofrem o processo reverso, geram imagens mentais interessantes. Elas apresentam, em cada linha, uma escala de gradientes. Pode-se interpretar esses gradientes como a distribuição de amostras treinadas que são menores ou iguais a um valor correspondente da escala. Isto foi fundamental para que se pudesse trabalhar com extração de regras em cima das redes WiSARD.

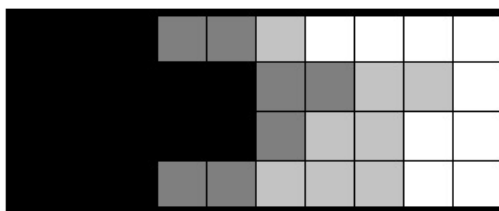


Figura 4.2: Exemplo de uma imagem mental de uma rede treinada com escalas termômetro

## 4.3 Algoritmos propostos

A algoritmo de extração de regras processa como entradas as imagens mentais e os valores originais dos elementos e suas respectivas classes e gera uma fila de candidatos à regra. Essas imagens mentais possuem como modelo a escala termômetro. Ele evoluiu do trabalho feito por Coutinho et. al., que analisava manualmente as imagens mentais para inferir as regras.

Obtidas as imagens mentais de cada discriminador, é feita uma comparação entre elas. Uma matriz de desvios padrões é criada, onde cada elemento da matriz representa o desvio padrão da posição correspondente entre as classes analisadas. Ordenam-se, então, esses elementos do maior para o menor desvio padrão. Optou-se por este critério pelo fato de que as melhores escolhas feitas manualmente também possuíam os maiores desvios padrões.

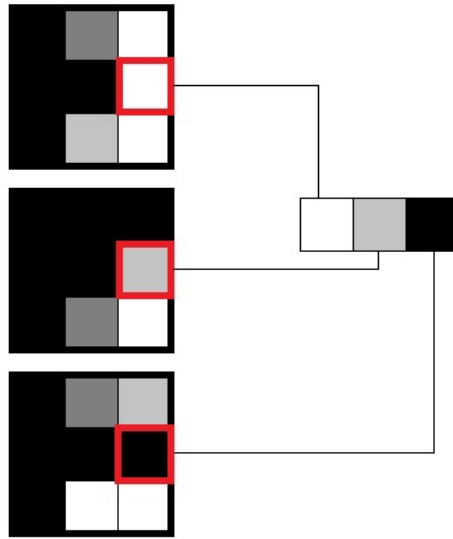


Figura 4.3: Comparação de valores entre imagens “mentais”

Em seguida, retira-se o valor do topo da fila. A linha do valor na matriz corresponde à variável da base de dados a ser comparada e a coluna, ao valor de corte da comparação. Esse valor de corte pode ser obtido pela seguinte fórmula:

$$valor_{corte} = minimo + \frac{maximo - minimo}{n_{escalas}} \times escala_{atual} \quad (4.2)$$

Tendo-se definido o valor de corte, é a vez de saber quais classes devem ser discriminadas. Calcula-se o valor médio de todas as amostras da classe por variável. Repete-se o mesmo para as demais classes. As classes que obtiverem médias abaixo do valor de corte são separadas das que tiverem médias acima deste. Optou-se pelas médias das variáveis por classe como uma maneira razoável mais simples de se discriminar.

A desigualdade ( $\leq$  ou  $>$ ) é definida como aquela que englobar mais classes. Se houver um número igual de classes discriminadas, é definido o valor  $\leq$  como padrão.

Feito isso tudo, essa condição é testada para todos os elementos treinados pelas redes. São retirados da lista de elementos todos aqueles que obedecerem à regra e que pertençam, de fato, ao conjunto de classes discriminadas pela regra, restando os elementos das demais classes e os da classe discriminada que não obedecem àquela condição (falsos). A matriz de desvios padrões é, então, recalculada, baseando-se nos exemplares remanescentes.

O processo, então, se repete até que restem elementos de uma única classe. Quando isso ocorrer, a árvore é completada considerando esses últimos elementos como não satisfatórios à última regra extraída.



Abaixo, encontra-se o algoritmo para a obtenção das condições:

---

**Algorithm 1** Algoritmo de extração de regras

---

```
função EXTRAIR_REGRAS(vetor_valores, vetor_classes, vetor_drasiw)
  cinza ← ESCALA_CINZA(vetor_drasiw) ▷ Transforma as imagens mentais
em matrizes em escala de cinza
  enquanto nClasses > 1 faça
    para i ← 1 até num_variaveis faça
      para j ← 1 até num_escalas faça
        TEMP_REGRA.ADICIONA(desvio_padrao(cinza(i,j))) ▷
Calcula o desvio padrão das amostras por escala e por varável e as inclui na lista
preliminar de regras.
      fim para
    fim para
    ORDENA_CRESCENTE(Temp_Regra,Desvio_padrao) ▷ Ordena em ordem
decrecente a fila de desvios padrões
    regra ← TEMP_REGRA.RETIRA()
    REGRAS.INCLUI(regra)
    CALCULA_VALOR(regra)
    para x ← 1 até vetor_valores faça
      se x ∈ Regra então
        vetor_valores.RETIRA(x)
        vetor_classes.RETIRA(classe_x)
      fim se
    fim para
    nClasses ← CALCULA_DISTINTOS(vetor_classes) ▷ Calcula quantas
classes distintas existem
  fim enquanto
fim função
```

---

O algoritmo funciona da seguinte forma:

- As imagens mentais de cada discriminador são comparadas e é gerada uma matriz de desvios padrões, correspondentes ao valores encontrados para cada classe..
- Escolhe-se a posição da matriz que apresenta o maior desvio padrão.
- A partir das coordenadas dessa matriz, acha-se a variável e o valor candidatos à regra de corte, que ajudarão na construção da regra.
- Compara-se esse valor candidato às médias das classes naquela posição para se achar a classe a ser discriminada. Todas as classes cujas médias estão abaixo desse valor são separadas das classes com médias acima desse valor. A desigualdade é então definida como aquela que menos satisfizer as classes.

- Com a matriz original de valores e o valor de corte, retiram-se do vetor de exemplares os valores classificados corretamente que pertencem à classe discriminada, restando os elementos das outras classes e aqueles classificados incorretamente (falsos positivos).
- Se todos os elementos de uma classe forem removidos, a matriz de desvios padrões é recalculada.

Esse algoritmo para quando existir apenas elementos pertencentes a apenas uma classe. Se isso ocorrer, a árvore é completada considerando esses últimos elementos como não satisfatórios à última regra extraída.

### 4.3.1 Construção das regras

Este algoritmo gera regras que procuram separar uma classe das demais. A separação das classes é definido pelos seguintes passos:

- Dada a variável candidata para comparação, calcula-se o valor médio de todos os elementos da classe pela variável. Repete-se o mesmo para as demais classes;
- Separam-se as classes que obtiverem médias abaixo do valor de corte das que tiverem médias acima deste;
- A desigualdade ( $\leq$  ou  $>$ ) é definida como aquela que englobar mais classes. Se houver um número igual de classes discriminadas, é definido o valor  $\leq$  como padrão.

Vamos supor, por exemplo, a condição  $k_i \leq 0,7$ . A caixa deve ser substituída por uma desigualdade. Vamos supor, também, três classes,  $C_1$ ,  $C_2$  e  $C_3$ . A tupla  $\{1, 2; 1, 0; 0, 5\}$  corresponde, respectivamente, à média dos elementos pertencentes a cada classe em relação à variável analisada. Sendo assim, a classe  $C_3$ , que possui média inferior ao valor de corte da condição, deve ser discriminada das outras duas classes,  $C_1$  e  $C_2$ , que possuem médias superiores. Logo, a condição ficará escrita dessa forma:

$$\text{Se } (k_i \leq 0,7) \text{ então } \textit{Classe} = C_3 \text{ senão } \dots \quad (4.3)$$

O algoritmo projetado para esta dissertação trabalha com bases de dados que tenham até três classes distintas, pois realiza a confecção da árvore de decisão apenas em profundidade, separando apenas uma classe das demais. Quando se tenta trabalhar com quatro classes, pode acontecer de uma mesma condição separar as classes duas a duas, fazendo-se necessário também um percurso em largura da árvore de decisão. Uma alternativa para este algoritmo poder trabalhar com mais classes

seria a utilização de recursão para percurso em largura combinado com percurso em profundidade.

# Capítulo 5

## Resultados e Discussão

Este capítulo mostra os passos subsequentes à extração das condições, que ainda não são regras formadas, no que diz respeito à elaboração de árvores de decisão. Nas seções seguintes, detalharemos os critérios e parâmetros utilizados para os testes do algoritmo desenvolvido.

### 5.1 Metodologia

Diferentemente do trabalho que deu origem a esta dissertação, optou-se por não adotar a técnica de *hill climbing*, já que houve pouca clareza na justificativa de uso do mesmo. Sendo assim, quanto ao número de escalas da rede, escolheram-se duas escalas múltiplas de 3 (3 e 6) e duas múltiplas de 5 (10 e 15). A escolha foi feita de modo arbitrário.

Todas as condições testadas utilizaram a configuração de 1 linha por entrada, já comprovada no artigo-base como sendo a melhor para que se extraia regras.

#### 5.1.1 Concatenações E e OU

Por vezes, mais de uma regra extraída separava um mesmo conjunto de classes de outro. Dessa forma, seria preciso que essas regras em questão fossem agrupadas de modo a melhor organizar a árvore de decisão. Assim, foram propostos neste trabalho dois tipos de concatenações: E e OU. Essas concatenações foram incluídos nos parâmetros de teste (referência cruzada) para se ter uma ideia de qual deles seria o mais adequado para o refinamento das regras extraídas. Não estão incluídas neste trabalho combinações mistas destes conectores, visto que precisariam de algoritmos mais complexos para definir essas combinações.

As regras geradas pelo algoritmo possuem o seguinte formato, considerando  $n$  como o número de condições extraídas que separam os mesmos conjuntos de classes:

$$\text{Se } (k_1 \leq v_1) \square (k_2 > v_2) \square \dots \square (k_n \leq v_n) \text{ então } Classe = C1 \text{ senão } \dots \quad (5.1)$$

Neste exemplo, os valores  $k$  seriam quaisquer características avaliadas da base de dados, os valores  $v$  seriam os valores de corte e as caixas, as combinações, que seriam substituídas por um dos dois conectores: E ou OU. A condição não satisfeita pode cair em uma nova regra ou uma classe diferente.

### 5.1.2 Rotina de testes

Com as bases de dados coletadas e tratadas, foram feitos os testes. Foi feita a referência cruzada dos seguintes parâmetros:

- Número de escalas: 3, 6, 10 e 15
- Tipo de regra: E ou OU

## 5.2 Resultados experimentais

A tabela abaixo mostra os resultados obtidos testando-se os dois algoritmos. O valor em negrito corresponde à configuração com a maior taxa de acerto encontrada.

Tabela 5.1: Taxa de acerto do algoritmo proposto

Base de Dados	Tipo de Regra	Número de escalas			
		3	6	10	15
<i>Iris</i>	E	<b>97,33%</b>	97,33%	90%	96,67%
	OU	64%	64%	66%	94,67%
Câncer de Mama	E	<b>94,42%</b>	81,12%	82,98%	87,55%
	OU	28,76%	30,62%	84,55%	86,27%
Vinhos	E	79,21%	<b>88,76%</b>	82,58%	87,64%
	OU	52,81%	55,62%	87,64%	60,67%

Um fato que deve ser levado em consideração é que toda vez que o algoritmo é executado com esses parâmetros, os resultados sempre serão os mesmos, ou seja, o desvio padrão é zero. Isto ocorre devido à arquitetura escolhida das redes WiSARD (1 RAM para cada *bit*), que produz os mesmos resultados para qualquer treinamento recebido com a mesma base de dados.

Com relação à classificação feita pela rede WiSARD, o algoritmo de extração de regras consegue ser melhor apenas com a base de dados *Iris*. A classificação

feita diretamente pela rede WiSARD é mais eficiente com as demais bases de dados. As comparações foram feitas utilizando-se as mesmas configurações de rede e os melhores resultados do algoritmo para cada base de dados.

Tabela 5.2: Comparação da taxa de acerto do algoritmo com a WiSARD

Base de Dados	Taxa de acerto	
	Imagens “mentais”	WiSARD
<i>Iris</i>	<b>97,33%</b>	95,33%
Câncer de Mama	94,42%	<b>95,71%</b>
Vinhos	88,76%	<b>95,51%</b>

### 5.2.1 Melhores resultados

Os melhores resultados dentre os avaliados foram encontrados com regras do tipo E e menos escalas. Esses resultados produziram as seguintes árvores de decisão:

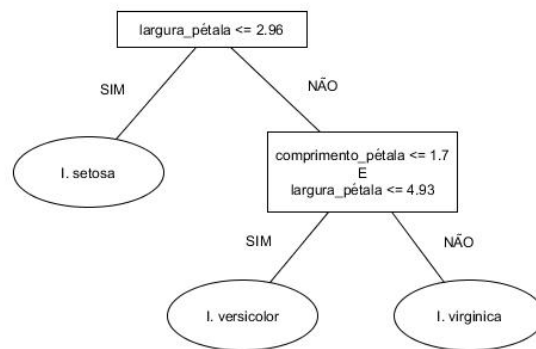


Figura 5.1: Árvore de decisão das flores Iris

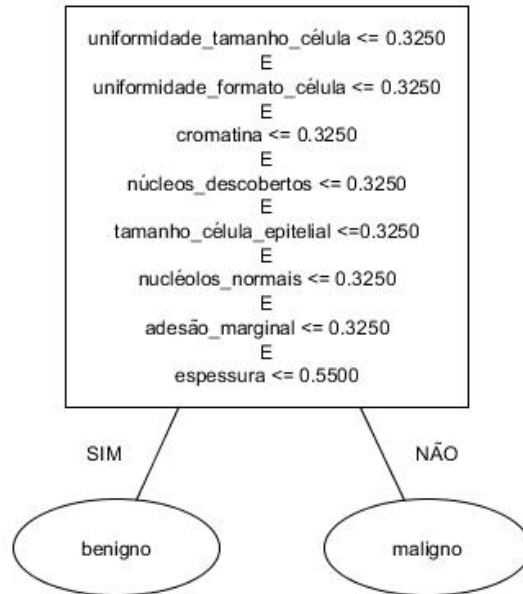


Figura 5.2: Árvore de decisão do câncer de mama

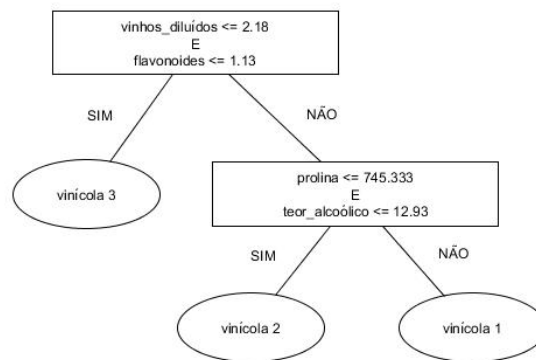


Figura 5.3: Árvore de decisão das vinícolas

### 5.3 Discussão

O algoritmo se saiu bem no geral ao interpretar as regras. Na base de dados das *Iris*, esteve bem próximo do estado da arte. Quanto à segunda base de dados, do câncer de mama, o algoritmo acabou utilizando muitas características para poder chegar ao resultado de 94,42%. Quanto à base de dados dos vinhos, utilizou-se um número de escalas maior, mas com poucas regras, para chegar a uma taxa de acerto de 88,76%.

### 5.3.1 Comparação com o estado da arte dos algoritmos

A tabela a seguir mostra uma comparação dos resultados de cada um dos algoritmos apresentados com o proposto neste trabalho.

Tabela 5.3: Taxa de acerto dos algoritmos

Base de Dados	Taxa de acerto					
	RxREN	Wang	G-REX	CAREX	Anomalias	Paulo
<i>Iris</i>	97,3%	98%	92%	96,67%	<b>99,46%</b>	97,33%
Câncer de Mama	96,4%	98,9%	96,2%	-	<b>99%</b>	94,42%
Vinhos	-	-	92,5%	<b>93,93%</b>	-	88,76%

Os resultados do teste se aproximam bastante do que é encontrado no estado da arte dos algoritmos, embora estejam um pouco abaixo destes no caso das bases de dados do câncer de mama e dos tipos de vinho. Os trabalhos que envolveram algoritmos do estado-da-arte não apresentaram as árvores de decisão resultantes da extração de regras.



# Capítulo 6

## Conclusão

Este trabalho em nenhum momento apresentou alguma intenção em tentar compreender o aprendizado de redes neurais WiSARD, como foi demonstrado em diversos trabalhos sobre extração de regras, pois as regras não foram extraídas diretamente da rede, mas das imagens “mentais”, que já apresentam certa compreensão do que foi aprendido.

Pode-se dizer, segundo a classificação de Andrews, que o algoritmo desenvolvido neste trabalho pertence à classe dos algoritmos pedagógicos, pois apenas se utiliza tratamento estatístico em cima das imagens mentais, que seriam as saídas da rede WiSARD, com a ajuda dos valores reais e das classes dos dados de entrada. Não há nenhuma manipulação direta do algoritmo na rede treinada.

O algoritmo projetado trabalha para diferenciar até três classes distintas. Para que se pudesse diferenciar mais classes, seria preciso aumentar a complexidade do algoritmo, percorrendo-se a árvore de decisão em profundidade para se criar regras em ambas as folhas de um nó, por exemplo.

Por vezes, as regras extraídas pelo algoritmo conseguem se sair melhor que a própria rede WiSARD, em que foram baseadas, para a tarefa de classificação, podendo ser utilizadas como complemento para tal rede. Isso não quer dizer, entretanto, que elas possam substituir a rede original treinada, como foi o caso da base de dados dos vinhos.

Apesar de os resultados estarem abaixo do estado da arte em extração de regras, este algoritmo mostra-se promissor nessa tarefa. Muito disso se deve pelo fato da utilização de desvios padrões como fator de escolha das regras, já que conseguiram reproduzir os resultados encontrados no trabalho em que esta dissertação se baseou.

Para os próximos passos desta pesquisa, consideram-se uma investigação mais profunda a respeito do tipo de concatenação a ser adotado na confecção das regras, incluindo concatenações mistas, a simplificação das regras extraídas e a geração de regras para problemas envolvendo mais classes.

# Referências Bibliográficas

- [1] GRIECO, B. P., LIMA, P. M., DE GREGORIO, M., et al. “Producing pattern examples from mental images”, *Neurocomputing*, v. 73, n. 7, pp. 1057–1064, 2010.
- [2] FU, L. “Rule generation from neural networks”, *Systems, Man and Cybernetics, IEEE Transactions on*, v. 24, n. 8, pp. 1114–1124, 1994.
- [3] ANDREWS, R., DIEDERICH, J., TICKLE, A. B. “Survey and critique of techniques for extracting rules from trained artificial neural networks”, *Knowledge-based systems*, v. 8, n. 6, pp. 373–389, 1995.
- [4] ZHOU, Z.-H. “Rule extraction: Using neural networks or for neural networks?” *Journal of Computer Science and Technology*, v. 19, n. 2, pp. 249–253, 2004.
- [5] AUGASTA, M. G., KATHIRVALAVAKUMAR, T. “Rule extraction from neural networksA comparative study”. In: *Pattern Recognition, Informatics and Medical Engineering (PRIME), 2012 International Conference on*, pp. 404–408. IEEE, 2012.
- [6] JIAN-GUO, W., JIAN-HONG, Y., WEN-XING, Z., et al. “Rule extraction from artificial neural network with optimized activation functions”. In: *Intelligent System and Knowledge Engineering, 2008. ISKE 2008. 3rd International Conference on*, v. 1, pp. 873–879. IEEE, 2008.
- [7] KONIG, R., JOHANSSON, U., NIKLASSON, L. “G-REX: A Versatile Framework for Evolutionary Data Mining”. In: *Data Mining Workshops, 2008. ICDMW’08. IEEE International Conference on*, pp. 971–974. IEEE, 2008.
- [8] MYSZKOWSKI, P. B. “Coevolutionary algorithm for rule induction”. In: *Computer Science and Information Technology (IMCSIT), Proceedings of the 2010 International Multiconference on*, pp. 73–79. IEEE, 2010.

- [9] H. HOU, G. D. “An Anomaly Detection-Based Classification System”. In: *2006 IEEE Congress on Evolutionary Computation*, Vancouver, BC, Canada, jul. 2006.
- [10] GRIECO, B. P., LIMA, P. M., DE GREGORIO, M., et al. “Extracting fuzzy rules from mental images generated by modified WiSARD perceptrons”. In: *Proc. E*, v. 26, pp. 101–773, 2008.
- [11] LUDERMIR, T. B., DE SOUTO, M. C., DE OLIVEIRA, W. R. “Weightless neural networks: knowledge-based inference system”. In: *Neural Networks, 2008. SBRN’08. 10th Brazilian Symposium on*, pp. 207–212. IEEE, 2008.
- [12] DE CERQUEIRA, E. O., DE ANDRADE, J. C., POPPI, R. J., et al. “Redes neurais e suas aplicações em calibração multivariada”, *Química Nova*, v. 24, n. 6, pp. 864–873, 2001.
- [13] MINSKY, M., PAPERT, S. “Perceptron: an introduction to computational geometry”, *The MIT Press, Cambridge, expanded edition*, v. 19, pp. 88, 1969.
- [14] ALEKSANDER, I., GREGORIO, M. D., FRANÇA, F. M. G., et al. “A brief introduction to Weightless Neural Systems”. In: *European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning*.
- [15] ALEKSANDER, I. “From WISARD to MAGNUS: a family of weightless virtual neural machines”, *RAM-Based Neural Networks*, v. 9, 1998.
- [16] KANERVA, P. “Sparse distributed memory and related models”, 1992.
- [17] ALEKSANDER, I., MORTON, H. “General neural unit: retrieval performance”, *Electronics letters*, v. 27, n. 19, pp. 1776–1778, 1991.
- [18] ALEKSANDER, I. “Logical connectionist systems”. In: *Neural computers*, Springer, pp. 189–197, 1989.
- [19] WICKERT, I., FRANÇA, F. M. “AUTOWISARD: Unsupervised Modes for the WISARD”. In: *Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence*, Springer, pp. 435–441, 2001.
- [20] SARKAR, B. K., SANA, S. S., CHAUDHURI, K. “A genetic algorithm-based rule extraction system”, *Applied soft computing*, v. 12, n. 1, pp. 238–254, 2012.

- [21] RODRÍGUEZ, M., ESCALANTE, D. M., PEREGRÍN, A. “Efficient distributed genetic algorithm for rule extraction”, *Applied soft computing*, v. 11, n. 1, pp. 733–743, 2011.
- [22] BOTZHEIM, J., HÁMORI, B., KÓCZY, L. T., et al. “Bacterial algorithm applied for fuzzy rule extraction”, 2002.
- [23] NÚÑEZ, H., ANGULO, C., CATALÀ, A. “Rule extraction from support vector machines.” In: *ESANN*, pp. 107–112, 2002.
- [24] MARTENS, D., BAESENS, B., VAN GESTEL, T., et al. “Comprehensible credit scoring models using rule extraction from support vector machines”, *European journal of operational research*, v. 183, n. 3, pp. 1466–1476, 2007.
- [25] ZHU, P., HU, Q. “Rule extraction from support vector machines based on consistent region covering reduction”, *Knowledge-Based Systems*, v. 42, pp. 1–8, 2013.
- [26] BAESENS, B., SETIONO, R., MUES, C., et al. “Using neural network rule extraction and decision tables for credit-risk evaluation”, *Management science*, v. 49, n. 3, pp. 312–329, 2003.
- [27] ISHIBUCHI, H., NAKASHIMA, T., MURATA, T. “Three-objective genetics-based machine learning for linguistic rule extraction”, *Information Sciences*, v. 136, n. 1, pp. 109–133, 2001.
- [28] HAYASHI, Y., SETIONO, R., YOSHIDA, K. “A comparison between two neural network rule extraction techniques for the diagnosis of hepatobiliary disorders”, *Artificial intelligence in Medicine*, v. 20, n. 3, pp. 205–216, 2000.
- [29] CRAVEN, M., SHAVLIK, J. “Rule extraction: Where do we go from here”, *University of Wisconsin Machine Learning Research Group Working Paper*, pp. 99–1, 1999.
- [30] TICKLE, A. B., ANDREWS, R., GOLEA, M., et al. “The truth will come to light: directions and challenges in extracting the knowledge embedded within trained artificial neural networks”, *Neural Networks, IEEE Transactions on*, v. 9, n. 6, pp. 1057–1068, 1998.
- [31] JOHANSSON, U., KONIG, R., NIKLASSON, L. “Automatically balancing accuracy and comprehensibility in predictive modeling”. In: *Information Fusion, 2005 8th International Conference on*, v. 2, pp. 7–pp. IEEE, 2005.

- [32] PAULO V. S. COUTINHO, HUGO C. C. CARNEIRO, D. S. C. F. M. G. F.  
“Extracting rules from DRASiWs mental images”. In: *European Symposium on Artificial Neural Networks - Advances in Computational Intelligence and Learning*.