

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

FACULDADE DE ENGENHARIA ELÉTRICA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA



**RECONHECIMENTO DE EDIFÍCIOS UTILIZANDO A
TRANSFORMADA DE GABOR, A TRANSFORMADA
WAVELET E A REDE NEURAL PERCEPTRON DE
MÚLTIPLAS CAMADAS**

LEANDRO AURELIANO DA SILVA

Uberlândia

2018

LEANDRO AURELIANO DA SILVA

**RECONHECIMENTO DE EDIFÍCIOS UTILIZANDO A
TRANSFORMADA DE GABOR, A TRANSFORMADA WAVELET E
A REDE NEURAL PERCEPTRON DE MÚLTIPLAS CAMADAS**

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal de Uberlândia, como requisito parcial para obtenção do título de **Doutor em Ciências**.

Área de Concentração: Engenharia Elétrica

Linha de Pesquisa: Processamento da Informação

Orientador: Prof. Dr. Gilberto Arantes Carrijo

Uberlândia

2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da UFU, MG, Brasil.

- S586r
2018
- Silva, Leandro Aureliano da, 1979-
Reconhecimento de edifícios utilizando a transformada de Gabor, a transformada Wavelet e a rede neural Perceptron de múltiplas camadas / Leandro Aureliano da Silva. - 2018.
118 f. : il.
- Orientador: Gilberto Arantes Carrijo.
Tese (Doutorado) - Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica.
Disponível em: <http://dx.doi.org/10.14393/ufu.te.2018.776>
Inclui bibliografia.
1. Engenharia elétrica - Teses. 2. Processamento de imagens - Teses. 3. Percepção geográfica - Teses. I. Carrijo, Gilberto Arantes. II. Universidade Federal de Uberlândia. Programa de Pós-Graduação em Engenharia Elétrica. IV. Título.

CDU: 621.3

LEANDRO AURELIANO DA SILVA

**RECONHECIMENTO DE EDIFÍCIOS UTILIZANDO A
TRANSFORMADA DE GABOR, A TRANSFORMADA WAVELET E
A REDE NEURAL PERCEPTRON DE MÚLTIPLAS CAMADAS**

Tese apresentada ao Programa de Pós-Graduação em
Engenharia Elétrica da Universidade Federal de
Uberlândia, como requisito parcial para obtenção do
título de **Doutor em Ciências**.

Área de Concentração: Engenharia Elétrica

Uberlândia, 15 de Maio de 2018

Banca Examinadora

Prof. Dr. Orientador Gilberto Arantes Carrijo, (UFU)

Prof. Dr. Antônio Cláudio Paschoarelli Veiga, (UFU)

Prof(a). Dr(a). Milena Bueno Pereira Carneiro, (UFU)

Prof. Dr. José Ricardo Gonçalves Manzan, (IFTM)

Prof. Dr. Luiz Fernando Batista Loja, (IFG)

AGRADEDIMENTOS

Primeiramente, agradeço a Deus por sempre transformar todas as dificuldades em aprendizados e conquistas.

A toda a minha família, em especial as minhas Mães, Ivone Maria da Silva e Joana Darc, ao meu Pai Ronaldo Rodrigues da Silva, a minha esposa Lucianna Camilo Vilela e aos meus filhos, Edson e Lucas pela compreensão, paciência e incentivo nos momentos mais difíceis.

Ao meu Pai Wilson, Célia e Irmãos João Vitor e Ana Clara pelo carinho.

Ao Professor Doutor Gilberto Arantes Carrijo pela confiança em conduzir a orientação deste trabalho.

Aos amigos Professores José Ricardo, Milena Bueno e Rodrigo Lemos, pelos conhecimentos transmitidos durante inúmeras discussões.

Aos amigos, George Kemil, William Gigo, Alcione Wagner, Romeu Pereira Abrahão, Eduardo Saad, Luciano Pimenta, Renato e todos os amigos da FACTHUS, instituição que trabalho e respeito muito pela amizade dos companheiros e colaboração.

A professora Maria Heliadora pelo apoio demonstrado para o desenvolvimento deste trabalho.

Aos grandes companheiros de Uberlândia Luiz, Jucelino, Cristiane e Eduardo pela amizade demonstrada ao longo dos anos.

A todos que contribuíram direta e indiretamente para o sucesso deste trabalho

Meus sinceros agradecimentos.

SUMÁRIO

LISTA DE FIGURAS	VII
LISTA DE TABELAS.....	X
LISTA DE SIGLAS E ABREVIATURAS.....	XII
ABSTRACT	XIV
CAPÍTULO 1.....	1
1. INTRODUÇÃO	1
1.1 - Introdução	1
1.2 - Motivação	3
1.3 - Trabalhos correlatos.....	4
1.4 - Objetivos deste trabalho.....	9
1.5 – Estrutura deste trabalho	10
1.6 - Considerações finais deste capítulo	11
CAPÍTULO 2.....	12
2. FILTRO DE GABOR E TRANSFORMADA WAVELET	12
2.1 - Introdução	12
2.2 - Filtro de Gabor	13
2.3 - Transformada Wavelet.....	18
2.3.1 - Transformada Wavelet como um banco de filtros	20
2.3.2 - Reconstrução da Wavelet.....	25
2.3.3 - Famílias de Wavelet.....	29
2.4 - Considerações finais deste capítulo	32
CAPÍTULO 3.....	33
3. REDES NEURAIS ARTIFICIAIS	33
3.1 - Introdução	33

3.2 - Introdução às redes neurais artificiais.....	34
3.3 - Breve histórico das redes neurais artificiais	34
3.4 - O Neurônio Biológico.....	36
3.5 - Modelo do neurônio artificial	37
3.5.1 - Funções de ativação	39
3.6 - Principais arquiteturas das redes neurais artificiais	43
3.7 - Tipos de treinamento das redes neurais artificiais	46
3.7.1 - Treinamento supervisionado	47
3.7.2 - Treinamento não supervisionado	47
3.8 - Perceptron camada única	47
3.9 - Método do gradiente descendente (regra delta).....	50
3.10 - Redes Perceptron múltiplas camadas.....	52
3.10.1 - Algoritmo backpropagation	54
3.11 - Considerações finais deste capítulo	67
CAPÍTULO 4.....	68
4. MÉTODO PROPOSTO	68
4.1 - Introdução	68
4.2 - Pré-Processamento	69
4.3 - Extração das características	70
4.4 - Redução da dimensionalidade	73
4.5 - Treinamento e teste da rede MLP	73
4.6 - Considerações finais deste capítulo	77
CAPÍTULO 5.....	78
5. RESULTADOS OBTIDOS	78
5.1 - Introdução	78
5.2 - Banco de dados	78
5.3 - Resultados obtidos	80
5.4 - Conclusões	95
CAPÍTULO 6.....	97

6. CONCLUSÕES GERAIS, CONTRIBUIÇÕES E TRABALHOS FUTUROS.....	97
6.1 - Introdução	97
6.2 - Conclusões	97
6.3 - Contribuições	98
6.4 - Trabalhos Futuros	99
6.5 - Considerações finais deste capítulo	99

LISTA DE FIGURAS

Figura 1.1 - (a) Imagens aéreas de edifícios; e (b) imagens obtidas de uma cidade. Fonte: Li, et al. 2014.....	2
Figura 2.1 - Filtragem de um estágio para a geração de aproximações (A) e detalhes (D) de um sinal (S). Fonte: Adaptado de Misiti et al. (2000).	21
Figura 2.2 - Decomposição da filtragem de um estágio sem e com sub-amostragem. Fonte: Adaptado de Misiti et al. (2000).	21
Figura 2.3 - Exemplo de uma árvore de decomposição até o nível 2.	22
Figura 2.4 - Reconstrução do sinal a partir dos coeficientes wavelet. Fonte: Adaptado de Misiti et al. (2000).	25
Figura 2.5 - Decomposição e reconstrução de um sinal. Fonte: Adaptado de Misiti et al. (2000).....	26
Figura 2.6 - Reconstrução do sinal a partir dos coeficientes de aproximação e de detalhe. Fonte: Adaptado de Misiti et al. (2000).....	27
Figura 2.7 - Reconstrução da aproximação A_1 a partir dos coeficientes cA_1 . Fonte: Adaptado de Misiti et al. (2000).	27
Figura 2.8 - Reconstrução da aproximação D_1 a partir dos coeficientes cD_1 . Fonte: Adaptado de Misiti et al. (2000).	27
Figura 2.9 - Componentes do sinal reconstruído. Fonte: Adaptado de Misiti et al. (2000).	28
Figura 2.10 - Forma da wavelet de Haar. Fonte: Misiti et al. (2000).	29
Figura 2.11 - Famílias de wavelets de Daubechies com as ordens variando de 2 a 10. Fonte: Misiti et al. (2000).....	30
Figura 2.12 - Família de wavelet coiflet para as ordens variando de 1 a 5. Fonte: Misiti et al. (2000).....	31
Figura 2.13 - Famílias de wavelets symlets para as ordens variando de 2 a 8. Fonte: Misiti et al. (2000).....	32

Figura 3.1 - Neurônio biológico e seus componentes. Fonte: (ALVES, 2007)	37
Figura 3.2 - Neurônio artificial. Fonte: (SILVA; SPATTI; FLAUZINO, 2010)	38
Figura 3.3 - Função degrau. Fonte: Adaptado (SILVA; SPATTI; FLAUZINO, 2010)	40
Figura 3.4 - Função sinal. Fonte: Adaptado (SILVA; SPATTI; FLAUZINO, 2010)	40
Figura 3.5 - Função rampa. Fonte: Adaptado (SILVA; SPATTI; FLAUZINO, 2010)	41
Figura 3.6 - Função logística. Fonte: Adaptado (SILVA; SPATTI; FLAUZINO, 2010)	42
Figura 3.7 - Função tangente hiperbólica. Fonte: Adaptado (SILVA; SPATTI; FLAUZINO, 2010)	42
Figura 3.8 - Rede <i>feedforward</i> . Fonte: Adaptado de Haykin (2007)	44
Figura 3.9 - Exemplo de uma rede <i>feedforward</i> de camadas múltiplas. Fonte: Adaptado de Haykin (2007)	45
Figura 3.10 - Exemplo de uma rede recorrente. Fonte: Adaptado de Haykin (2007)	46
Figura 3.11 - Perceptron. Fonte: (SILVA; SPATTI; FLAUZINO, 2010)	48
Figura 3.12 - Exemplo de uma rede MLP com duas camadas ocultas. Fonte: Cassimiro (2013)	53
Figura 3.13 - Notação para a derivação do algoritmo <i>backpropagation</i> . Fonte: Cassimiro (2013)	55
Figura 4.1 - Notação para a derivação do algoritmo <i>backpropagation</i> . Fonte: Cassimiro (2013)	55
Figura 4.2- (a) imagem original; e (b) imagem em níveis de cinza	70
Figura 4.3 - Módulo do resultado obtido da convolução entre a imagem em níveis de cinza e a representação de wavelet Gabor	71

Figura 5.1 - (a) imagens da primeira categoria, (b) imagens da terceira categoria, (c) imagens da sexta categoria, (d) imagens da décima sexta categoria, (e) imagens da vigésima categoria de edifícios. Fonte: http://eeepro.shef.ac.uk/building/dataset.rar	79
Figura 5.2 - Gráfico comparativo entre cA3 e cD3 para Daubechies com a ordem variando de 1 a 10.	86
Figura 5.3 - Gráfico comparativo entre cA3 e cD3 para Coiflet com a ordem variando de 1 a 5.	87
Figura 5.4 - Gráfico comparativo entre cA3 e cD3 para Symlet com a ordem variando de 1 a 10.	87
Figura 5.5 - Gráfico comparativo entre cA3 e cD3 com redução do número de características.	92

LISTA DE TABELAS

Tabela 5.1- Recursos computacionais usados nesse trabalho.....	80
Tabela 5.2 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Daubechies usando os coeficientes de aproximação de nível 3.....	81
Tabela 5.3 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Coiflet usando os coeficientes de aproximação de nível 3.	82
Tabela 5.4 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Symlet usando os coeficientes de aproximação de nível 3.	83
Tabela 5.5 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Daubechies usando os coeficientes de detalhe de nível 3.....	84
Tabela 5.6 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Coiflets usando os coeficientes de detalhe de nível 3.	85
Tabela 5.7 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Symlets usando os coeficientes de detalhe de nível 3.....	85
Tabela 5.8 - Taxa de reconhecimento da MLP variando o parâmetro de orientação do filtro de Gabor para TW de Daubechies de ordem 5 usando o coeficiente de aproximação.....	88
Tabela 5.9 - Taxa de reconhecimento da MLP variando o parâmetro de orientação do filtro de Gabor para TW de Daubechies de ordem 5 usando o coeficiente de detalhe.....	89
Tabela 5.10 - Taxa de reconhecimento da MLP variando o parâmetro de orientação do filtro de Gabor para TW de Daubechies de ordem 5 usando o coeficiente de aproximação com fator de redução de linha e coluna igual a 10.	90
Tabela 5.11 - Taxa de reconhecimento da MLP variando o parâmetro de orientação do filtro de Gabor para TW de Daubechies de ordem 5 usando o coeficiente de detalhe com fator de redução de linha e coluna igual a 10.....	91

Tabela 5.12 - Tempos de processamentos gastos no treinamento e no teste da rede neural MLP para o melhor resultado usando o coeficiente de aproximação de nível 3.	93
Tabela 5.13 - Tempos de processamentos gastos no treinamento e no teste da rede neural MLP para o melhor resultado usando o coeficiente de detalhe de nível 3.	93
Tabela 5.14 - Resultado obtido utilizando o método proposto nesse trabalho e os resultados encontrados na literatura.	94

LISTA DE SIGLAS E ABREVIATURAS

- SFBR - Reconhecimento de edifícios baseado em filtros orientados; expressão traduzida do Inglês “*Steerable filter-based building recognition*”.
- PCA - Análise de componentes principais; expressão traduzida do Inglês “*Principal component analysis*”.
- LDA - Análise discriminante linear; expressão traduzida do Inglês “*Linear discriminant analysis*”.
- SDA - Análise discriminante semi-supervisionado; expressão traduzida do Inglês “*Semi-supervised discriminant analysis*”.
- SLPP - Preservação da projeção local (método supervisionado); expressão traduzida do Inglês “*Locality preserving projections supervised*”.
- TW - Transformada Wavelet.
- TDW - Transformada Discreta Wavelet.
- TDIW - Transformada Discreta Inversa Wavelet.
- MLP - Perceptron de Múltiplas camadas, do inglês “*Multi Layer Perceptron*”.
- RNA - Rede Neural Artificial.
- MRA - Análise de multiresolução, do inglês “*Multi-Resolution Analysis*”.
- VBO - Vetores Bipolares Ortogonais.

RESUMO

O reconhecimento de edifícios é uma tarefa bastante difícil, pois as imagens utilizadas possuem ângulos e condições de iluminação diferentes, obstruções parciais de árvores, veículos em movimento ou outros edifícios. Devido a todas essas dificuldades, esse trabalho propõe um sistema de reconhecimento que utiliza a representação de wavelet Gabor para a extração de características, a transformada wavelet para a redução de dimensionalidade e para o reconhecimento a rede neural Perceptron de múltiplas camadas. Para verificar a eficiência deste sistema, utilizou-se um banco de dados de imagens de edifícios de prédios novos e antigos. O desempenho do método proposto foi satisfatório em relação a outros métodos encontrados na literatura.

Palavras Chaves: Reconhecimento de edifícios, Representação de wavelet Gabor, Extração de características e rede Perceptron de múltiplas camadas.

ABSTRACT

Recognition of buildings is a very difficult task, because the images used have different angles and lighting conditions, partial obstructions of trees, moving vehicles or other buildings. Due to all these difficulties, this paper proposes a recognition system which uses Gabor wavelet representation for the extraction of features, the wavelet transform for dimensionality reduction and the multi-layer perceptron neural network for recognition. In order to check the efficiency of this system, we used a database of images of new and old buildings. The performance of the proposed method was satisfactory compared to other methods found in the literature.

Keywords: Recognition of buildings, Gabor wavelet representation, extraction of features and multi-layer perceptron

CAPÍTULO 1

1. INTRODUÇÃO

1.1 - Introdução

O reconhecimento de edifícios destina-se a reconhecer diferentes edifícios em uma grande escala de imagens. A identificação dos edifícios pode contribuir para várias aplicações, tais como: detecção automática de alvo, localização de robô em tempo real, navegação visual, sistemas de segurança pública, projeto arquitetônico e reconstrução 3D de cidades e até mesmo integração com o google earth.

Por exemplo, esse reconhecimento permite que robôs se localizem e naveguem facilmente. Portanto, é útil para a localização do robô e navegação. Além disso, permite que um soldado com um simples toque no celular, seja capaz de identificar o alvo para uma determinada explosão. No entanto, possibilita que designers a partir de uma base de dados na internet pesquisem modelos de construção desejáveis.

As imagens utilizadas para o reconhecimento de edifícios podem ser obtidas de duas formas: aéreas e de edifícios em cidades. Como mostrado na Figura 1.1, estes dois tipos de imagens são fundamentalmente diferentes em aparência.

Figura 1.1 - (a) Imagens aéreas de edifícios; e (b) imagens obtidas de uma cidade. Fonte: Li, et al. 2014.



As imagens aéreas são obtidas sobrepondo as fotografias digitais capturadas por diferentes câmeras de vídeo, sensores equipados em avião ou uma plataforma de alta altitude capaz de sobrevoar uma cidade (*drone*). Esses equipamentos são geralmente de alta-resolução e podem ser influenciados pela velocidade, altitude da aeronave, condições climáticas, tipo de dados e métodos de pós-processamento aplicado. Além disso, para o reconhecimento dos edifícios essas imagens requerem um elevado custo computacional e uma grande quantidade de armazenamento dos dados.

As imagens de edifícios capturadas de uma cidade podem ser obtidas por câmeras digitais ou vídeo. Devido a variabilidade de parâmetros existentes nessas imagens, o seu reconhecimento é mais difícil.

O reconhecimento de edifícios pode ser considerado como um reconhecimento de objetos ou, para uma determinada categoria específica, uma recuperação de imagens baseada em conteúdo.

As imagens de edifícios obtidas de uma cidade apresentam uma variabilidade de parâmetros. Como por exemplo, essas imagens podem ser capturadas em diferentes ângulos, diversas condições de iluminação ou podem ocorrer obstruções parciais de árvores, de veículos em movimento ou outros

edifícios. Devido a essa variabilidade, as imagens dos edifícios apresentam diferentes transformações geométricas e fotométricas. Assim, o sistema de reconhecimento de edifícios deve ser robusto a essas transformações.

A proposta deste trabalho é desenvolver um sistema de reconhecimento de edifícios que utiliza a representação de wavelet Gabor para extrair as características, a transformada wavelet para reduzir a dimensionalidade dessas características e a rede Perceptron de múltiplas camadas para o reconhecimento dos edifícios, tendo como característica fundamental para esse reconhecimento apenas a textura.

Esse capítulo apresenta a motivação, os trabalhos correlatos, os objetivos e a estrutura deste trabalho. Finalmente, são realizadas as considerações finais deste capítulo.

1.2 - Motivação

Na sociedade de hoje a tecnologia móvel desempenha um papel importante. No entanto, uma parcela cada vez mais crescente possui agregados aos seus celulares uma unidade de processamento cada vez mais rápida acompanhada por uma câmara digital de alta resolução.

Assim, surgiu a motivação de desenvolver um sistema de reconhecimento de edifícios robusto às características fotométricas e geométricas. Esse sistema possibilitaria que com apenas um toque no celular fosse possível utilizá-lo. As aplicações para esse tipo de reconhecimento são diversas. Pode-se citar: turismo, navegação móvel, reconhecimento de alvo, arquitetura, dentre outras.

1.3 - Trabalhos correlatos

Segundo (LI e ALLINSON, 2013), as três etapas de um sistema de reconhecimento de edifícios utilizando imagens capturadas em cidades são: a representação das características; as características correspondentes e a classificação.

A representação das características é uma das etapas mais importantes do sistema de reconhecimento, pois ela influencia diretamente no desempenho do classificador. O objetivo dessa representação é extrair diferentes objetos de uma imagem. Essa extração pode ser realizada de forma global ou local.

As características globais são extraídas dos pixels de uma imagem. Essas características são a cor, a forma, a textura, dentre outras.

As características locais são definidas pela repetição de padrões de uma variação local dos pixels. Essas características tendem a ser invariantes às transformações geométricas e fotométricas.

Após extrair as características de cada imagem, essas são agrupadas com a finalidade de encontrar no banco de dados pares de imagens correspondentes (clusterização). A métrica utilizada para essa finalidade pode ser a distância euclidiana ou a distância de Mahalanobis, dentre outras (LI, et al. 2014) e (PEDRINI e SCHWARTZ, 2008). Finalmente, a classificação é realizada para determinar o melhor par de imagens que corresponde a imagem de teste.

O reconhecimento de edifícios chamou a atenção de pesquisadores que utilizam imagens aéreas (ZHANG, 2013) e imagens de edifícios de uma cidade (LI, et al. 2014).

Mayer (1999) apresentou uma revisão do estado da arte de extração automática de imagens aéreas de edifícios. No entanto, não existem pesquisas dessas imagens capturadas de cidade.

Segundo Li e Allinson (2013) existem três métodos que utilizam imagens de edifícios capturadas de uma cidade. Esses métodos são: 1) agrupamento

(*clustering*); 2) algoritmos que utilizam representação de características e 3) outros, ou seja, sistemas que não utilizam nem o método 1 nem o 2 ou sistemas que combinam esses dois métodos.

No método que utiliza o agrupamento, nas várias estruturas da imagem são extraídas hierarquicamente as relações semânticas entre as diferentes características visuais de baixo nível. Por exemplo, podem ser utilizados segmentos de linha reta na recuperação de imagens baseada em conteúdo (CBIR) dos edifícios.

Em suas pesquisas, Li e Shapiro (2002) utilizaram para cada segmento de linha da imagem a cor, a orientação e a informação espacial. Esses segmentos foram integrados e agrupados em um tipo de característica de nível médio originando os *clusters* de linhas consistentes. Assim, as relações *intracluster* e *intercluster* foram usadas para reconhecer diferentes edifícios.

Zhang e Kosecká (2007) propuseram o reconhecimento de edifícios hierárquico (HBR). O HBR é baseado na detecção do ponto de fuga e nos histogramas de cores localizado. Nesse sistema os segmentos de linha detectados são agrupados em direções dominantes de fuga e os pontos de fuga são estimados pelo algoritmo de máxima expectância (EM). Assim, é utilizado um limiar da magnitude do gradiente do pixel da imagem e o resultado é um dos grupos denominados de: esquerdo, direito e vertical. Se a diferença entre sua direção gradiente e as direções dos pontos de fugas principais é menor do que um limiar, então é calculado um histograma de cores localizado apenas nestes pixels e o resultado obtido é indexado sobre um dos vetores definidos como esquerdo, direito e vertical. Finalmente, os histogramas são casados pela distância chi-quadrado e os resultados do reconhecimento são extraídos da base de dados ZuBud (<http://www.vision.ee.ethz.ch/showroom/zubud/>) utilizando o descritor *Scale-invariant feature* (SIFT) (LOWE, 2004). Após, aplica-se nestes resultados um modelo probabilístico simples para integrar a evidência do casamento individual. Devido a indexação usando histogramas de cores

localizados, este método é relativamente eficaz e tem motivado os pesquisadores.

Li et al. (2014) afirmaram que o sistema desenvolvido por Zhang e Kosecká possui limitações. Essas limitações são: (1) eles assumiram que em cada imagem a ser reconhecida só existe um edifício. No entanto, em cenas reais nem sempre isso acontece; (2) embora eles tenham utilizado um método de indexação rápido, o tempo de processamento ainda é longo para a extração de muitas características das imagens a cores; e (3) o desempenho do algoritmo no reconhecimento só é satisfatório quando o fundo da imagem é simples. Entretanto, o algoritmo é inadequado para sistemas de navegação que necessitam do processamento em tempo real.

No trabalho de Duan e Allinson (2010) foi desenvolvido um novo método para detecção automática de pontos de fuga e agrupamento de linhas associadas a fachadas de uma única imagem de edifício não calibrada. A detecção destes pontos é precisa na reconstrução desse tipo de imagem que foi retificada e reconstruída em 3D. Eles utilizaram nos testes o banco de imagens ZuBud (<http://www.vision.ee.ethz.ch/showroom/zubud/>). Os resultados obtidos nos testes mostraram que o método proposto por Duan e Allinson (2010) é eficiente para vários tipos de estruturas de edifícios e condições onde as imagens são filtradas pelos pontos de fuga.

Chung, Han e He (2009) utilizaram as representações baseadas no esboço dos principais componentes estruturais de um edifício para encontrar suas janelas e portas. Para obter essas componentes eles utilizaram um detector de regiões invariantes baseado em linhas de nível conhecido como *Maximally stable extremal region* (MSER). O resultado do MSER é normalizado usando o histograma de gradientes orientados. Depois, Chung et al, utilizaram a técnica *k-means* para o agrupamento dos diferentes componentes estruturais e espectrais. Assim, foi possível que o agrupamento de uma imagem consultada encontrasse uma correspondência com o agrupamento de uma imagem da base de dados.

Este método foi projetado especialmente para o reconhecimento de edifícios de imagens que possuem grandes mudanças de ponto de vista. É importante ressaltar que eles não realizaram testes, por exemplo, em condições de oclusão.

Em sistemas que utilizam algoritmos para representar as características locais, Hutchings e Mayol (2005) desenvolveram um sistema de reconhecimento de edifícios para dispositivos móveis. Esse sistema é utilizado como um guia de turismo. Por exemplo, para uma imagem de consulta, suas características locais são extraídas e descritas respectivamente pelo detector de canto de Harris e pelo descritor *scale invariant feature transform* (SIFT). No processo correspondente (reconhecimento), para cada imagem de consulta conforme a sua posição no sistema de posição global (GPS) é selecionada uma escala. Isso resulta em uma redução do espaço de busca e em um menor custo computacional. No entanto, esse sistema falha quando existem grandes mudanças em relação a diversos pontos de vista.

O Groeneweg et al. (2006) implementaram um método denominado de reconhecimento rápido *offline*. Esse método utiliza a detecção da região de intensidade e a análise de componentes principais (PCA). O método de Groeneweg et al. utiliza uma elipse ao redor de cada região detectada, redimensionando e representando cada uma delas nas suas cores usando o sistema RGB. Após a normalização pela soma das intensidades de todos os pixels na região, a técnica de PCA é aplicada e as características são agrupadas em *clusters*. Cada *cluster* é caracterizado pelo seu centroide. Então, para a medida de distância é adotada uma votação majoritária ponderada. Esta abordagem em um telefone móvel reduz o custo computacional e a capacidade de armazenamento, mas é sensível às cores de iluminação e não é invariante à rotação.

Li e Allinson (2009) desenvolveram um sistema de reconhecimento que utiliza extração de características visuais. Eles também analisaram como os diferentes algoritmos de redução de dimensionalidade influenciam nesse

reconhecimento. Os algoritmos utilizados por Li e Allisson na redução de dimensionalidade foram: análise de componentes principais (PCA), análise de discriminante linear (LDA), análise de discriminante semi-supervisionado (SDA), preservação da projeção local (do inglês: *locality preserving projections*). Essa preservação pode ser um método supervisionado, como por exemplo, *supervised locality preserving projections (SLPP)* e um método não supervisionado, pode-se citar *unsupervised locality preserving projections (LPP)*.

Trinh et al. (2008) propuseram um método para reconhecer vários edifícios. Nesse método, as fachadas de cada edifício são extraídas considerando os segmentos de linha e a detecção de pontos de fuga. Em seguida, aplica-se o descritor SIFT e o resultado desse descritor é utilizado para descrever cada edifício.

Li e Allinson (2013) propuseram um modelo de reconhecimento simples, denominado reconhecimento de edifícios usando filtros orientados (SFBR). Apesar desses filtros serem utilizados em outros trabalhos, foi a primeira vez que eles foram aplicados no reconhecimento de edifícios. As quatro partes do modelo SFBR são: I) a representação de características; II) a seleção dessas características; III) a redução de dimensionalidade; e IV) a classificação.

Na representação das características Li e Allinson (2013) usaram os filtros orientados que possibilitaram a extração de características locais em diferentes pontos de vista. Na seleção dessas características foi utilizada a técnica de *pooling* e na redução da dimensionalidade à análise de discriminante linear. Finalmente, para o reconhecimento eles usaram a máquina de vetores de suporte (SVM).

1.4 - Objetivos deste trabalho

O objetivo deste trabalho é desenvolver um sistema de reconhecimento de edifícios capaz de resolver os problemas da grande variabilidade dos parâmetros fotométricos e geométricos das imagens utilizadas nesse reconhecimento.

Os objetivos específicos deste trabalho são:

- Com a finalidade de preservar as características fotométricas e geométricas da imagem do edifício, aplicar o filtro de Gabor na extração de características dessa imagem, usando a textura como principal característica;
- Utilizar a transformada discreta wavelet na redução de dimensionalidade do vetor de características obtido após a aplicação do filtro de Gabor na imagem;
- Usar a rede Perceptron de múltiplas camadas no treinamento e teste para o reconhecimento dos edifícios;
- Verificar o comportamento da rede Perceptron de múltiplas camadas variando as ordens e a família da transformada discreta wavelet utilizada; e
- Analisar o comportamento da rede Perceptron de múltiplas camadas, variando o número de orientações do filtro de Gabor.

1.5 – Estrutura deste trabalho

Este trabalho está estruturado em 6 capítulos:

Este capítulo apresenta a motivação, os trabalhos correlatos, os objetivos e a estrutura deste trabalho. Finalmente, são realizadas as considerações finais deste capítulo.

O Capítulo 2 descreve o filtro de Gabor e a transformada wavelet. Finalmente, são realizadas as considerações finais deste capítulo.

O Capítulo 3 apresenta uma rápida introdução às redes neurais artificiais. Em seguida é apresentado um breve histórico sobre essas redes, o neurônio biológico, o modelo do neurônio artificial, as principais arquiteturas de redes neurais artificiais e os tipos de treinamentos destas redes. Depois, é descrito o Perceptron camada única, o método do gradiente descendente (regra delta) e a rede Perceptron de múltiplas camadas (MLP) que será utilizada nesse trabalho. E finalmente, são realizadas as considerações finais deste capítulo.

O Capítulo 4 descreve o método proposto neste trabalho. E finalmente, são realizadas as considerações finais deste capítulo.

O Capítulo 5 mostra o banco de dados utilizados neste trabalho e apresenta os resultados obtidos nos testes realizados. Finalmente, são realizadas as conclusões destes resultados.

O Capítulo 6 apresenta as conclusões e as contribuições deste trabalho e os trabalhos futuros que poderão serem desenvolvidos a partir dessa tese. Finalmente, são realizadas as considerações finais deste capítulo.

1.6 - Considerações finais deste capítulo

Esse capítulo apresentou a motivação, os trabalhos correlatos, os objetivos e a estrutura deste trabalho.

O próximo capítulo descreve o filtro de Gabor e a transformada wavelet. Finalmente, são realizadas as considerações finais deste capítulo.

CAPÍTULO 2

2. *FILTRO DE GABOR E TRANSFORMADA WAVELET*

2.1 - Introdução

Uma etapa muito importante na classificação de imagens é a extração de características. A finalidade dessa extração é reduzir a grande quantidade de informações para processar os dados e assim classificá-los.

Segundo Cook e Wolf (1998) apud Pasquali et al. (2007) existem três categorias de extração de características. A primeira contém as características espectrais do sinal ou da imagem a ser analisada. Essas características estão relacionadas com a energia emitida ou refletida por esse sinal ou por essa imagem. Na segunda categoria as características estão relacionadas à morfologia matemática que é aplicada à região ou à vizinhança dos pixels de uma imagem. O objetivo da terceira é extrair as características espaciais descritas pela forma, dimensão e textura da imagem que está sendo estudada.

Um dos atributos importantes usados na classificação de imagens e utilizado nesse trabalho é a textura. Para extrair a textura nas imagens usadas nessa tese utilizou-se o filtro de Gabor. Este filtro extrai diferentes tipos de texturas presentes nas imagens descritas pela frequência e pela orientação já definidas em suas funções.

Após extrair as características das imagens é necessário reduzir a dimensionalidade destas informações. Para essa redução, esse trabalho utiliza a sub-amostragem da transformada wavelet discreta.

Esse capítulo descreve o filtro de Gabor e a transformada wavelet. Finalmente, são realizadas as considerações finais deste capítulo.

2.2 - Filtro de Gabor

Em 1946, Denis Gabor utilizou a desigualdade de Schwarz e a incerteza de Heisenberg para mostrar que a caracterização de um sinal simultaneamente nos domínios temporal e das frequências espaciais apresentam um limite inferior. Como mostrado na Equação (2.1) esse limite é definido pelo produto de sua extensão pela sua duração (ANGELO, 2000).

$$\Delta t \cdot \Delta \omega \geq \frac{1}{4\pi} \quad (2.1)$$

Baseado nos trabalhos de Heisenberg e Weyl, Gabor propôs uma família de funções capaz de alcançar o limite teórico inferior de incerteza conjunta nos domínios temporal e das frequências. Essas funções ficaram conhecidas como filtros de Gabor ou representação de Wavelet Gabor. A Equação (2.2) mostra a formulação desenvolvida por Gabor.

$$g(x, \sigma, k) = \exp \left\{ -\frac{1}{2} \left(\frac{x}{\sigma} \right)^2 + 2\pi i k x \right\} \quad (2.2)$$

onde:

$g(x, \sigma, k)$ – filtro de Gabor;

σ – extensão do envelope gaussiano que limita espacialmente o filtro de Gabor;

e

k – frequência da onda senoidal associada ao filtro.

Nos últimos anos tem sido aplicado o filtro de Gabor em imagens principalmente na extração dos atributos de textura. Como por exemplo, em reconhecimento de face, reconhecimento de íris, reconhecimento de impressão digital, entre outras.

Em 1985, Daugman desenvolveu o filtro de Gabor em duas dimensões. As funções de Gabor expandidas para duas dimensões simulam muito bem o comportamento dos perfis de campos receptivos encontrados nas células do córtex cerebral de mamíferos. Estas células capturam as propriedades de localização espacial, orientação, frequência espacial, entre outras (ANGELO, 2000).

Seguindo a heurística de Piparsaniyan, Sharma e Mahapatra (2014) e Struc e Pavesic (2009), os filtros de Gabor em duas dimensões são definidos pela Equação (2.3).

$$\left. \begin{aligned} \psi_{\Pi}(f, \theta, \gamma, \eta) &= \frac{f^2}{\pi \gamma \eta} e^{-\left(\frac{f^2}{\gamma^2} x_t^2 + \frac{f^2}{\eta^2} y_t^2 \right)} e^{j 2 \pi f x_t} \\ x_t &= x \cdot \cos(\theta) + y \cdot \sin(\theta) \\ y_t &= -x \cdot \sin(\theta) + y \cdot \cos(\theta) \end{aligned} \right\} \quad (2.3)$$

onde:

$$j = \sqrt{-1},$$

f – frequência senoidal;

θ – orientação da wavelet;

γ – comprimento da onda senoidal;

η – comprimento de onda do espaço wavelet que é perpendicular à onda senoidal;

(x, y) – coordenada dos pixels;

$\psi_{\Pi}(f, \theta, \gamma, \eta)$ – banco de filtros da representação wavelet Gabor;

x_t – parâmetros referentes as orientações; e

y_t – parâmetros referentes as orientações.

Como mostrado na Equação (2.4), variando-se a frequência (f) e a orientação (θ) na Equação (2.3), obtém-se o banco de filtros da representação de wavelet Gabor ($\psi_{g,h}(x, y)$) (Piparsaniyan, Sharma e Mahapatra (2014)).

$$\left. \begin{aligned} f &= \frac{f_{\max}}{(\sqrt{2})^g} \\ \theta &= \frac{h}{8} \pi \\ \psi_{g,h}(x, y) &= \psi_{\Pi}(f, \theta, \gamma, \eta) \end{aligned} \right\} \quad (2.4)$$

onde:

f – frequência;

θ – orientação da wavelet; e

$\psi_{g,h}(x, y)$ – banco de filtros da representação de wavelet Gabor.

Para obter o banco de filtros da representação de wavelet Gabor, considera-se na Equação (2.4) $\gamma = \eta = \sqrt{2}$, $f_{\max} = 0,25$ ciclo por pixel, o intervalo de variação da escala de frequência (g) variando de 0 a 4 e o intervalo

de variação das orientações (θ) variando de 0 a 7 (LIU, et al., 2010) (PIPARSANIYAN; SHARMA; MAHAPATRA, 2014).

Como mostrado na Equação (2.5), convoluindo a imagem original com o banco de filtros da representação em duas dimensões da wavelet Gabor, obtém-se a imagem filtrada.

$$O_{g,h}(x,y) = I(x,y) * \psi_{g,h}(x,y) \quad (2.5)$$

onde:

$I(x,y)$ – imagem original;

$\psi_{g,h}(x,y)$ – banco de filtros da representação de wavelet Gabor; e

$O_{g,h}(x,y)$ – imagem filtrada.

O resultado obtido na Equação (2.5), origina sub-imagens com 5 escalas de frequências e 8 orientações (PIPARSANIYAN; SHARMA; MAHAPATRA, 2014).

Após filtrar a imagem utilizando a Equação (2.5), obtém-se o vetor de características (F) que será utilizado para a classificação das imagens. Esse vetor é encontrado sub-amostrando com um fator igual a 4 as linhas e as colunas do módulo da imagem filtrada. O resultado dessa operação é mostrado na Equação (2.6) (PIPARSANIYAN; SHARMA; MAHAPATRA, 2014).

$$F = (O_{0,0}^T \ O_{0,1}^T \ \cdots \ O_{4,7}^T) \quad (2.6)$$

onde:

F – vetor de características utilizado na classificação das imagens; e

$O_{g,h}(x,y)$ – imagem filtrada.

Segundo Piparsaniyan, Sharma e Mahapatra (2014) e Struc e Pavesic (2009) após obter o vetor de características, ele deve ser normalizado com média e variância iguais a zero e um, respectivamente. Para uma imagem $m \times n$, a dimensão (d) de F é mostrada na Equação (2.7).

$$d = \frac{m \cdot n \cdot g \cdot h}{4 \times 4} \quad (2.7)$$

onde:

m e n – número de linhas e colunas da imagem original, respectivamente;

g – variação da escala de frequência; e

h – variação das orientações.

Como mostrado na Equação (2.3) é importante ressaltar que as funções utilizadas para a representação de wavelet Gabor são senóides complexas bidimensionais modeladas por uma função Gaussiana bidimensional. O objetivo dessas funções é extrair as texturas encontradas nas imagens descritas pela frequência (f) e pela orientação (θ). Ao aplicar em uma imagem a Equação (2.3) consegue-se extrair características nessa imagem que apresentam diferentes texturas. Nesse trabalho aplica-se essa equação nas imagens dos edifícios para obter as diferentes texturas dessas imagens.

2.3 - Transformada Wavelet

Nos últimos anos a transformada wavelet (TW) foi utilizada largamente em diversas aplicações. Algumas dessas aplicações foram codificação em sub-bandas, análise de imagens, análise de redução de ruído em sinais de voz (SILVA; et al. 2014), análise de multiresolução, modelagem de sistemas variantes no espaço-tempo, compressão de vídeo, reconhecimento de objetos, dentre outras (OLIVEIRA, 2007). Ao aplicar a TW na análise de imagens e utilizar uma escala menor de imagens é possível representar toda a imagem.

As wavelets desenvolvidas por Haar ficaram no anonimato por muitos anos. E por muito tempo, elas foram a única base ortonormal de wavelets conhecida (SOARES, 2009).

Segundo Oliveira (2007) as transformadas de wavelet foram desenvolvidas da análise de sinais de ondas acústicas sísmicas realizadas por Morlet, Grossman, Meyer, Battle, Lemarié, Cohen, Mallat, Coifman, Rioul, entre outros.

O conceito de wavelet só foi utilizado no início da década de 80 por Alex Grossmann da Université de Marseille e Jean P. Morlet (Elf Aquitaine) quando eles analisaram sinais geofísicos usados na exploração de petróleo e gás.

A TW é aplicada nas mais diversas áreas da ciência. Em 1987, Stéphane Mallat desenvolveu o algoritmo piramidais em um único termo incorporando a teoria dessa transformada em processamento digital de sinais (WEEKS, 2012).

Meyer construiu a primeira wavelet não-trivial (suave) baseada no trabalho de Mallat (OLIVEIRA, 2007). Ao contrário das wavelets de Haar, que não são continuamente diferenciáveis, o que de certo modo limita suas aplicações, as wavelets de Meyer são continuamente diferenciáveis. No entanto, essas funções não têm suporte compacto.

Ingrid Daubechies também baseou no trabalho de Mallat e construiu o conjunto de wavelets ortogonais de suporte compacto mais utilizado. As

wavelets desenvolvidas por ela, tornaram-se um marco nas aplicações de processamento de sinais discretos. Os trabalhos de Daubechies são a base das aplicações atuais na área de wavelets.

Conforme Misiti et. al. (2000), a Equação (2.8) define o procedimento de análise para a transformada wavelet contínua.

$$Wf(a,b) = \int_{-\infty}^{\infty} f(t) \psi_{a,b}(t) dt \quad (2.8)$$

onde:

$Wf(a, b)$ – transformada wavelet;

$f(t)$ – sinal no tempo; e

$\psi_{a,b}(t)$ – núcleo da transformada wavelet.

Como mostrado na Equação (2.9), para um sinal discreto de N pontos, a integral da Equação (2.8) pode ser aproximada por um somatório. O resultado dessa aproximação é a transformada discreta wavelet (TDW).

$$Wf(a,b) = \sum_{n=0}^{N-1} f(n) \psi_{a,b}(n) \quad (2.9)$$

onde:

$Wf(a, b)$ – transformada wavelet;

$f(n)$ – sinal discreto no tempo; e

$\psi_{a,b}(n)$ – núcleo da transformada wavelet de tempo discreto.

A função $\psi_{a,b}(n)$, denominada wavelet, é derivada da função $\psi(t)$ chamada de “wavelet mãe” pela transformação mostrada na Equação (2.10).

$$\psi_{a,b}(n) = \frac{1}{\sqrt{a}} \psi\left(\frac{n-b}{a}\right) \quad (2.10)$$

onde:

$\psi\left(\frac{t-b}{a}\right)$ – núcleo da transformação;

b – posição ou a translação da wavelet; e

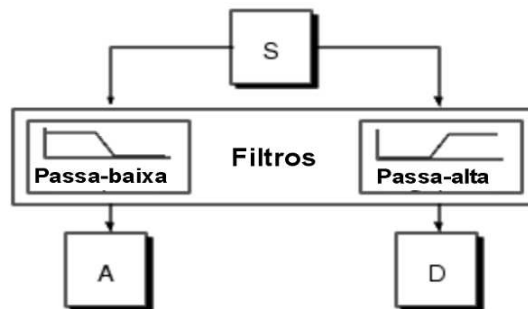
a – parâmetro de escala que está associado à largura da janela.

Existe uma ampla gama de escolhas para a wavelet mãe. Dentre elas, pode-se citar: Haar, Daubechies, symlets, coiflet e várias outras (OLIVEIRA, 2007). Cada uma possui as suas particularidades em relação a suavidade e a compactação.

2.3.1 - Transformada Wavelet como um banco de filtros

Em muitas aplicações a transformada discreta wavelet (TDW) é utilizada como um banco de filtros. Essa operação de filtragem é realizada usando dois filtros. O filtro passa-baixa origina os coeficientes de aproximações (A) que correspondem ao conteúdo de baixa frequência do sinal. E o filtro passa-alta origina os coeficientes de detalhes (D) que correspondem a alta frequência do sinal. A Figura 2.1 mostra essa decomposição.

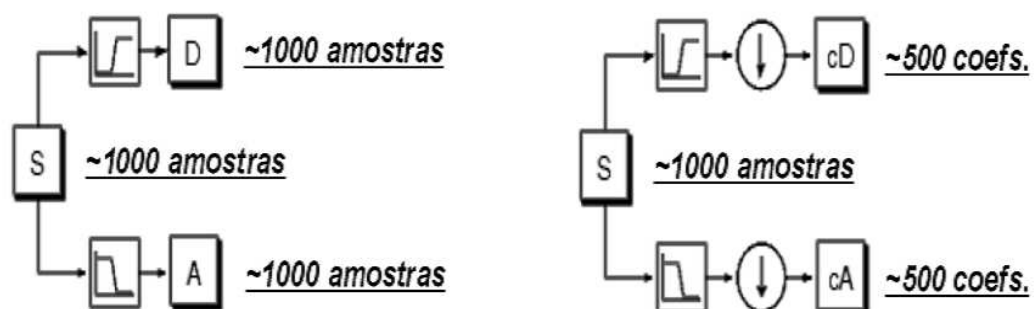
Figura 2.1 - Filtragem de um estágio para a geração de aproximações (A) e detalhes (D) de um sinal (S). Fonte: Adaptado de Misiti et al. (2000).



Da forma como está mostrado na Figura 2.1, o resultado da TDW apresenta duas vezes mais dados do que o sinal original. Isso diminui a eficiência computacional. Para resolver esse problema, a decomposição é realizada reduzindo a taxa de amostragem (*downsampling*) na mesma proporção que o número de decomposições. Nessa figura, esse número é igual a dois.

Ao utilizar a decomposição usando a *downsampling* é possível reduzir a dimensionalidade do vetor de características obtido pela filtragem de Gabor. A Figura 2.2 mostra essa decomposição.

Figura 2.2 - Decomposição da filtragem de um estágio sem e com sub-amostragem. Fonte: Adaptado de Misiti et al. (2000).



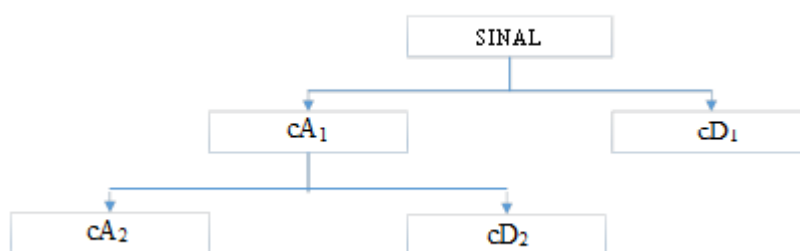
S – sinal;

cA – coeficiente de aproximação que corresponde a informação de baixa frequência do sinal (S); e

cD – coeficiente de detalhe que corresponde a informação de alta frequência do sinal (S).

A decomposição do sinal (S) usando a TDW pode ser repetida diversas vezes, mas sempre utilizando o coeficiente de aproximação (cA). Essa repetição origina a chamada árvore de decomposição wavelet. A Figura 2.3 ilustra um exemplo dessa árvore até o nível 2.

Figura 2.3 - Exemplo de uma árvore de decomposição até o nível 2.



A árvore de decomposição wavelet pode ser realizada até que o coeficiente de detalhe corresponda a uma única amostra (MISITI et al. 2000).

A árvore de decomposição usando a TDW na forma de um banco de filtros, só é possível pela análise de multiresolução ou do inglês “*Multi-Resolution Analysis* (MRA)”. Assim, para que seja possível a decomposição dos coeficientes usando a wavelet é necessário a aplicação de uma função chamada de escalonamento (*Scaling function*) ($\phi(t)$). Para encontrar essa função, Riccioti (2006) descreveu a funcionalidade de realizar esse escalonamento utilizando equações.

Conforme a heurística de Riccioti (2006), a função de escalonamento $\int \phi(t)$ deve ser contínua, geralmente ela é real como mostrado na Equação (2.11) e deve satisfazer a condição de admissibilidade.

$$\int \phi(t) \cdot dt = 1 \quad (2.11)$$

Como ilustrado na Equação (2.12), para construir a base das funções de escalonamento, $\phi(t)$ é deslocada por valores discretos e em cada nível de resolução é dilatada por um fator de escala (2^n).

$$\phi(t) = \sqrt{2} \sum_{n \in \mathbb{Z}} h_d(n) \phi(2t - n) \quad (2.12)$$

onde:

$\phi(t)$ – função de escalonamento com os fatores inteiros de escala (n); e

$h_d(n)$ – coeficiente do filtro passa-baixa que deve satisfazer a Equação (2.13).

$$\sum_{n \in \mathbb{Z}} h_d(n) = 1 \quad (2.13)$$

Segundo Ferreira (2009) e Oliveira (2007) pode-se calcular a wavelet utilizando a MRA. A Equação (2.14) mostra esse cálculo.

$$\psi(t) = \sqrt{2} \sum_{n \in \mathbb{Z}} g_d(n) \phi(2t - n) \quad (2.14)$$

onde:

$\psi(t)$ – wavelet mãe;

$g_d(n)$ – coeficientes do filtro passa-alta;

$\phi(2t - n)$ – função de escalonamento; e

n – quantidade de valores positivos discretos da escala.

Assim, os coeficientes de aproximação (cA) e detalhe (cD) são obtidos utilizando as Equações (2.15) e (2.16), respectivamente.

$$cA_j(n) = \sum_s f(s)h_d(2k-s) \quad (2.15)$$

$$cD_j(n) = \sum_s f(s)g_d(2k-s) \quad (2.16)$$

onde:

$f(s)$ – sinal a ser decomposto;

h_d e g_d – filtros de meia banda;

j – nível de decomposição;

cA_j e cD_j – coeficientes de aproximação e detalhe obtidos pela TWD; e

s – fator de discretização.

Segundo Oliveira (2007), os coeficientes $h_d(n)$ e $g_d(n)$ podem ser encontrados utilizando as Equações (2.17) e (2.18), respectivamente.

$$h_d(n) = \sqrt{2} \int_{-\infty}^{\infty} \phi(t)\phi(2t-n)dt \quad (2.17)$$

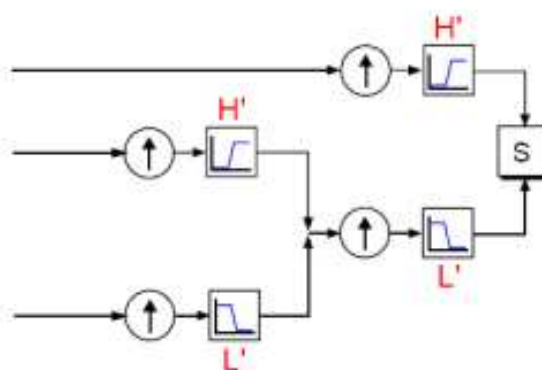
$$g_d(n) = \sqrt{2} \int_{-\infty}^{\infty} \psi(t)\phi(2t-n)dt \quad (2.18)$$

Em suma a análise de multiresolução consiste em dividir o espectro de um sinal em sub-bandas e então de acordo com o objetivo pretendido tratar individualmente cada sub-banda.

2.3.2 - Reconstrução da Wavelet

Como mostrado na sub-seção 2.3.1, a TDW decompõe o sinal em sub-bandas. Entretanto, para obter novamente o sinal original sem perda de informação, utiliza-se um processo conhecido como reconstrução ou síntese. A manipulação matemática aplicada nesse processo é a transformada discreta inversa de Wavelet (TDIW). A reconstrução do sinal pode ser realizada a partir dos coeficientes de wavelet. A Figura 2.4 ilustra esse procedimento.

Figura 2.4 - Reconstrução do sinal a partir dos coeficientes wavelet.
Fonte: Adaptado de Misiti et al. (2000).

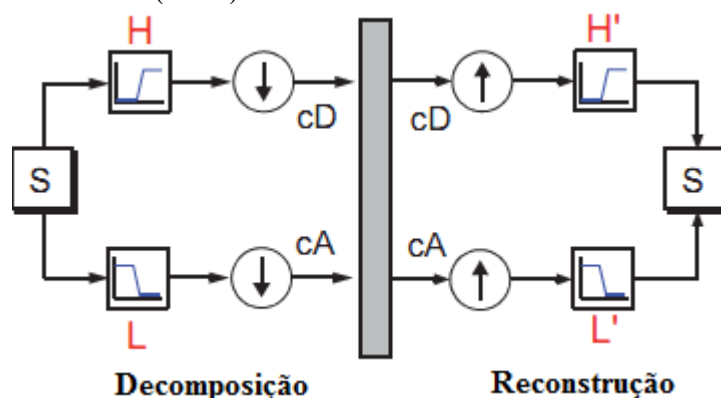


Com a aplicação da TDW no sinal, devido a filtragem e a sub-amostragem (*downsample*), as amostras são reduzidas por um fator de 2. A reconstrução é realizada usando a TDIW. A TDIW é aplicada nos coeficientes de aproximação e detalhe obtidos pela TDW. Essa inversa é uma operação de filtragem de reconstrução e sobre-amostragem (*upsampling*). O processo de *upsampling* consiste no alongamento do sinal pela inserção de zeros entre as amostras de forma que o sinal tenha o mesmo tamanho do sinal original.

Para uma perfeita reconstrução do sinal original é muito importante a escolha do filtro de reconstrução. Devido a diminuição do número de amostras do sinal, é introduzido nele distorções (*aliasing*). A escolha correta do filtro de

reconstrução pode cancelar essas distorções. A Figura 2.5 mostra como o sinal original pode ser recuperado.

Figura 2.5 - Decomposição e reconstrução de um sinal. Fonte: Adaptado de Misiti et al. (2000).



Como mostrado na Figura 2.6, é possível verificar a reconstrução das aproximações e dos detalhes a partir dos vetores de coeficientes.

Como ilustrado na Figura 2.7, para obter A_1 a partir do vetor de coeficientes cA_1 utiliza-se o filtro de reconstrução.

É importante observar na Figura 2.7 que o vetor dos coeficientes cD_1 é substituído por um vetor de zeros.

O tamanho do coeficiente de aproximação A_1 obtido na reconstrução é igual ao tamanho do sinal original S . Esse coeficiente é uma real aproximação deste sinal. De forma análoga, o coeficiente de detalhe D_1 pode ser obtido pelo mesmo processo de obtenção do coeficiente de aproximação. A Figura 2.8 ilustra a obtenção do coeficiente de detalhe.

Figura 2.6 - Reconstrução do sinal a partir dos coeficientes de aproximação e de detalhe. Fonte: Adaptado de Misiti et al. (2000).

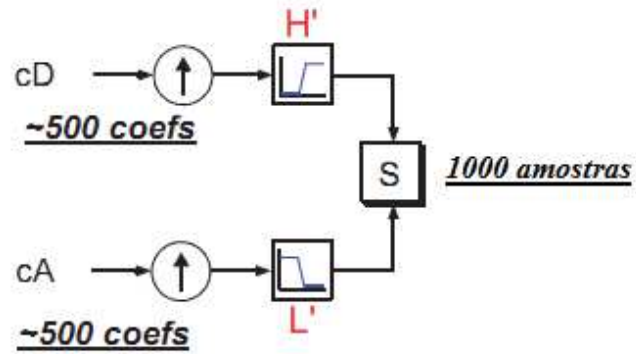


Figura 2.7 - Reconstrução da aproximação A_1 a partir dos coeficientes cA_1 . Fonte: Adaptado de Misiti et al. (2000).

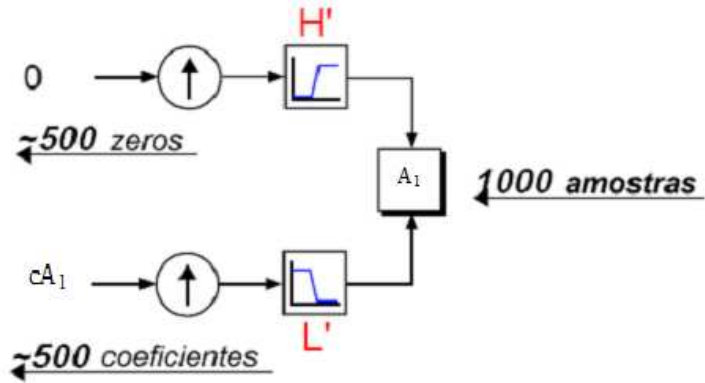
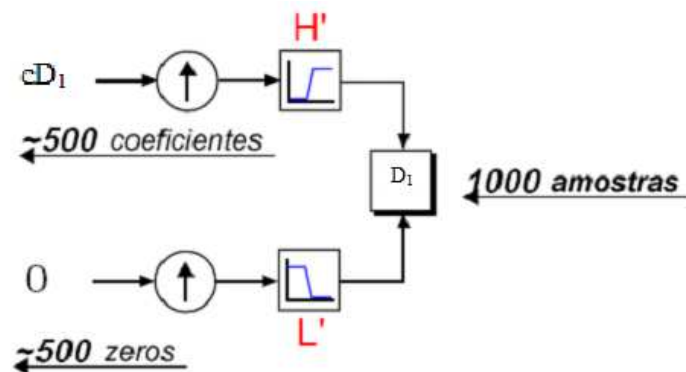


Figura 2.8 - Reconstrução da aproximação D_1 a partir dos coeficientes cD_1 . Fonte: Adaptado de Misiti et al. (2000).



Como mostrado na Equação (2.19), os coeficientes de detalhe e de aproximação são combinados para obter o sinal S .

$$S = A_1 + D_1 \quad (2.19)$$

onde:

S – sinal;

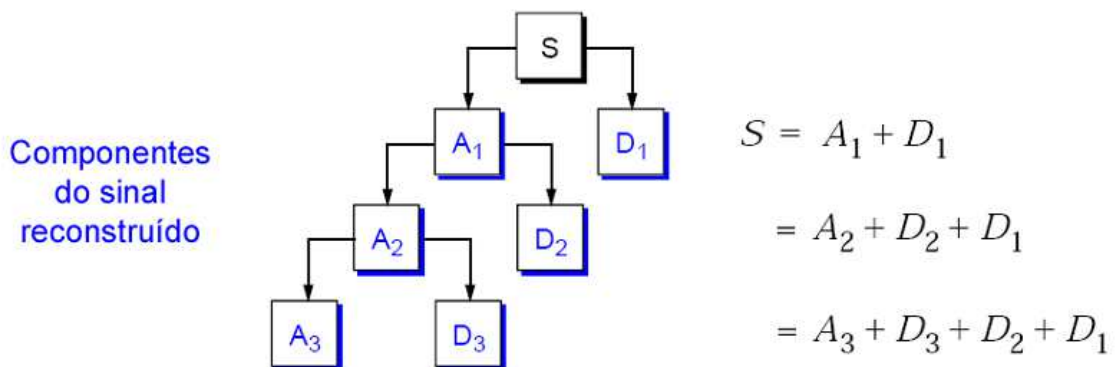
A_1 – coeficiente de aproximação; e

D_1 – coeficiente de detalhe.

É importante ressaltar que os vetores de coeficientes cA_1 e cD_1 não podem ser utilizados diretamente para reconstruir o sinal, pois eles apresentam a metade do comprimento do sinal original. Por isso, é necessário a reconstrução de aproximações e detalhes.

Na análise usando multi-nível, podem ser utilizadas todas as relações mostradas anteriormente neste item para todos os constituintes do sinal reconstruído. A Figura 2.9 mostra esse processo.

Figura 2.9 - Componentes do sinal reconstruído. Fonte: Adaptado de Misiti et al. (2000).



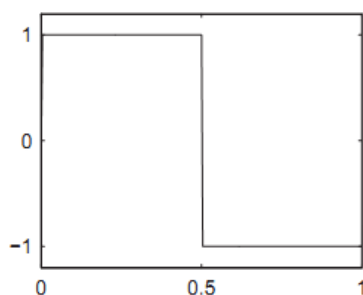
2.3.3 - Famílias de Wavelet

A análise de multiresolução desenvolvida por Mallat em 1989 foi responsável em fornecer subsídios teóricos para diversos pesquisadores desenvolverem a sua própria família de wavelets (OLIVEIRA, 2007). As quatro famílias de wavelet apresentadas nesse trabalho são: Haar, Daubechies, Coiflet e Symlet.

2.3.3.1 - Wavelet de Haar

A wavelet de Haar possui a forma da função degrau. Essa wavelet é compacta, mas não é continuamente diferenciável. Isso dificulta sua aplicação. A Figura 2.10 ilustra a forma da wavelet de Harr.

Figura 2.10 - Forma da wavelet de Haar. Fonte: Misiti et al. (2000).



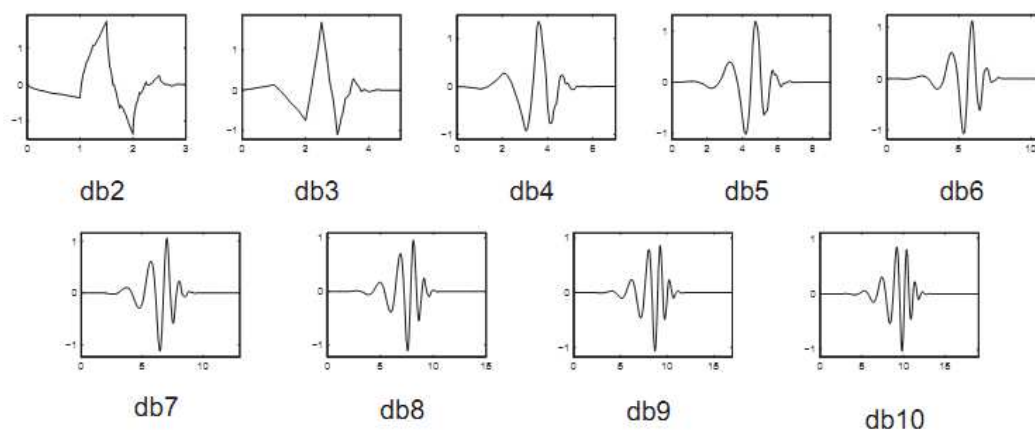
2.3.3.2 - Wavelets de Daubechies

As wavelets de Daubechies foram propostas por Ingrid Daubechies. Elas são famílias de wavelets ortogonais e de suporte compacto que possibilitaram a utilização da transformada discreta wavelet. Essa família também é

caracterizada por um número máximo de momentos nulos. Para cada tipo de wavelet desta classe existe uma função de escala chamada de wavelet pai. Essa função origina uma análise de multiresolução ortogonal.

Os nomes da família de Daubechies são escritos como dbN , onde N representa a ordem da função. A wavelet $db1$ é a wavelet de Haar. A Figura 2.11 mostra as funções de Daubechies para as ordens de 2 a 10 (MISITI et al. 2000).

Figura 2.11 - Famílias de wavelets de Daubechies com as ordens variando de 2 a 10. Fonte: Misiti et al. (2000).



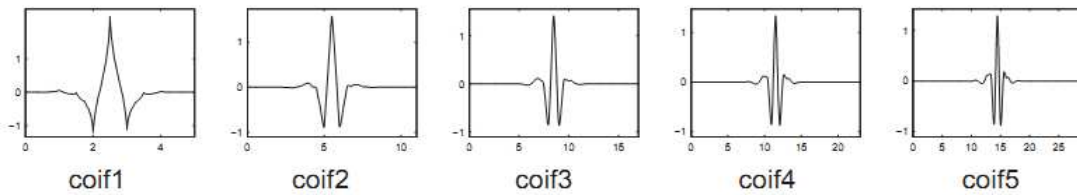
2.3.3.3 - Wavelets tipo Coiflets

A pedido de Ronaldo Coifman, a wavelet coiflets foi proposta por Ingrid Daubechies. Essa função wavelet e a função escala possuem cada uma $2N$ momentos iguais a 0. O comprimento máximo de cada uma dessas duas funções pode ser $6N-1$.

O objetivo da construção da wavelet coiflet foi garantir uma semelhança maior entre o sinal original e o sinal obtido aplicando essa transformada. A Figura 2.12 ilustra as wavelets coiflets para as ordens variando de 1 a 5. A

wavelet mãe dessa transformada é denominada coifN , onde N representa a ordem da wavelet (MISITI et al. 2000).

Figura 2.12 - Família de wavelet coiflet para as ordens variando de 1 a 5. Fonte: Misiti et al. (2000).

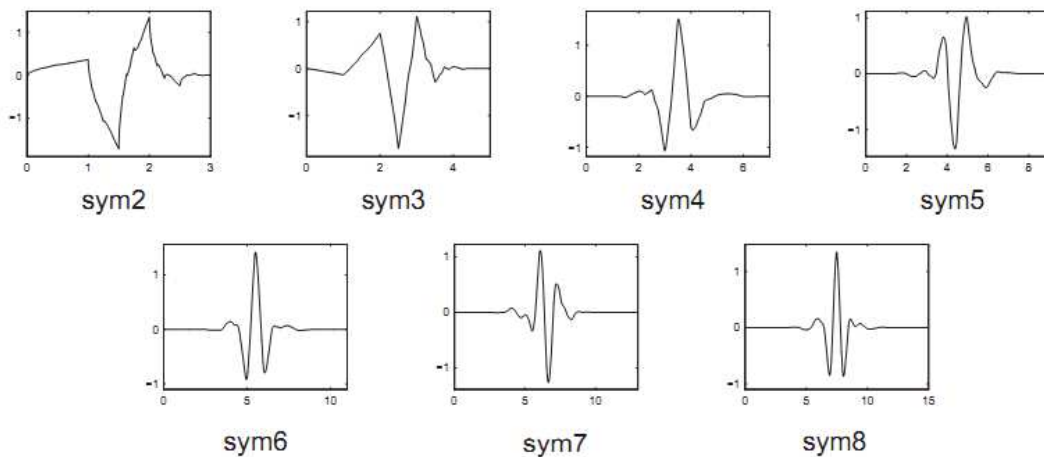


2.3.3.4 - Wavelets tipo Symlets

Como uma modificação da família db, as wavelets symlets foram propostas por Daubechies. Segundo Misiti. et. al. (2000), as symlets são wavelets quase simétricas e as propriedades das famílias de Daubechies e symlets são similares.

Segundo Ferreira (2009), quando comparada à família de Daubechies as symlets, apresentam suporte compacto com menor número de assimetria e maior número de decaimento. A Figura 2.13 mostra a família de symlets para as ordens variando de 2 a 8.

Figura 2.13 - Famílias de wavelets symlets para as ordens variando de 2 a 8. Fonte: Misiti et al. (2000).



2.4 - Considerações finais deste capítulo

Esse capítulo descreveu o filtro de Gabor e a transformada wavelet. Esse filtro é utilizado para extrair as características das imagens dos edifícios usadas nesse trabalho. E a transformada wavelet é utilizada para reduzir a dimensionalidade do vetor de características obtido após passar a imagem do edifício por este filtro.

O próximo capítulo apresenta uma rápida introdução às redes neurais artificiais. Em seguida é apresentado um breve histórico sobre essas redes, o neurônio biológico, o modelo do neurônio artificial, as principais arquiteturas de redes neurais artificiais e os tipos de treinamento dessas redes. Depois, é descrito o Perceptron camada única, o método do gradiente descendente (regra delta) e a rede Perceptron de múltiplas camadas que será utilizada nesse trabalho. E finalmente, são realizadas as considerações finais deste capítulo.

CAPÍTULO 3

3. *REDES NEURAIS ARTIFICIAIS*

3.1 - Introdução

As redes neurais artificiais foram desenvolvidas do desejo de representar o processo de aprendizagem do cérebro humano por meio de programas computacionais. Existem diversas aplicações nas quais essas redes podem ser empregadas. Pode-se citar: reconhecimento de caracteres, reconhecimento de imagens, análise de risco de crédito, detecção de fraudes, detecção de falhas em sistemas industriais, agrupamento da sequência do DNA, mineração de dados, previsão do tempo, previsão financeira, modelagem de sistemas dinâmicos, previsão de sequências de DNA, dentre outras (BRAGA; CARVALHO; LUDEMIR, 2007).

O objetivo deste capítulo é apresentar uma breve contextualização sobre as redes neurais artificiais. Esse capítulo inicia com uma rápida introdução às redes neurais artificiais. Em seguida é apresentado um breve histórico sobre essas redes, o neurônio biológico, o modelo do neurônio artificial, as principais arquiteturas de redes neurais artificiais e os tipos de treinamento dessas redes. Depois, é descrito o Perceptron camada única, o método do gradiente descendente (regra delta) e a rede Perceptron de múltiplas camadas que será utilizada nesse trabalho. E finalmente, são realizadas as considerações finais deste capítulo.

3.2 - Introdução às redes neurais artificiais

As redes neurais artificiais (RNA's) também chamadas de connexionismo ou sistema de processamento paralelo e distribuído são modelos computacionais que se assemelham ao sistema nervoso biológico (BRAGA; CARVALHO; LUDEMIR, 2007). Esses modelos são apresentados como uma estrutura computacional capaz de aprender usando exemplos e de generalizar a informação. Essa capacidade é obtida por meio de um conjunto de elementos de processamento denominados de neurônios artificiais que são interconectados pelas sinápses artificiais.

3.3 - Breve histórico das redes neurais artificiais

Na década de 40 iniciou a história das redes neurais artificiais, mais precisamente no ano de 1943 quando McCulloch e Pitts propuseram o modelo matemático do neurônio artificial que se assemelhava ao neurônio biológico (SILVA; SPATTI; FLAUZINO, 2010). O modelo desenvolvido por esses autores foi de um neurônio com processamento binário capaz de realizar muitas operações lógicas. Essa estrutura embora fosse muito simples foi responsável pelo desenvolvimento dos primeiros computadores digitais e dos modelos matemáticos que se assemelhavam às analogias biológicas (HAYKIN, 2007).

Em 1949, o neurofisiologista Donald Hebb apresentou o primeiro método de aprendizagem usado em redes neurais lançando o livro “*The organization of behavior*” (HEBB, 1949). Ele preconizou que a intensidade das conexões sinápticas é alterada em função dos erros localmente detectáveis fazendo com que os sistemas nervosos complexos, possam ser reduzidos a sistemas de processamento individual (KOVÁCS, 1996). No início essa teoria desenvolvida

por ele, teve uma boa aceitação entre os psicólogos, porém não teve quase nenhum impacto na comunidade de engenharia (HAYKIN, 2007).

Segundo (SILVA; SPATTI; FLAUZINO, 2010), em 1958 Frank Rosenblatt construiu o modelo básico do Perceptron e desenvolveu o primeiro neurocomputador chamado de Mark – I Perceptron. Esse modelo utilizava uma rede de múltiplos neurônios do tipo discriminadores lineares. O fundamento destes discriminadores foram as linhas de pensamento de McCulloch para desenvolver o modelo matemático de sinapse humana.

Em 1960 Widrow & Hoff desenvolveram o algoritmo do mínimo quadrado médio (*Least Mean Square – LMS*) que foi utilizado para formular a rede *Adaptive Linear Element* (Adaline) (HAYKIN, 2007). Posteriormente, eles propuseram a associação de múltiplas redes Adalines, resultando na rede Madaline, cujo aprendizado é fundamentado no algoritmo *LMS* ou regra delta (FAUSSET, 1994).

Nesse momento, vários pesquisadores entusiasmados por esses trabalhos iniciaram várias pesquisas nessa área. Porém, em 1969, Minsky e Papert publicaram o livro “*Perceptrons – an introduction to computational geometry*” onde demonstraram a limitação das redes neurais de camada única como o Perceptron e a Adaline, em resolver problemas bem simples como a função lógica Xor (ou-exclusivo). Em suma, eles demonstraram que essas redes não classificavam problemas onde as classes não eram linearmente separáveis (SILVA; SPATTI; FLAUZINO, 2010).

Após a obra idealizada por Minsky e Papert, houve um período com pouquíssimas pesquisas nessa área, com exceção dos pesquisadores que trabalhavam com a neurociência e a psicologia (Haykin, 2007). Algumas dessas pesquisas merecem destaque, dentre elas: o algoritmo de predição utilizando gradiente reverso obtido por Werbos em 1974, a implementação da rede *Adaptive Resonance Theory* (ART) desenvolvida por Grossberg em 1980,

a formulação dos mapas auto-organizáveis por Kohonen em 1982 e neste mesmo ano as redes recorrentes desenvolvidas por Hopfield.

O trabalho realizado por Hopfield fez com que as RNA's ocupassem novamente um lugar de destaque que possuía antes de 1969 (SILVA; SPATTI; FLAUZINO, 2010). A retomada definitiva dessa área deve-se a diversos fatores tais como: o desenvolvimento de computadores com poder de processamento maior, os algoritmos otimizados eficientes e robustos e as novas descobertas sobre o sistema nervoso. Assim, a teoria de redes neurais só conseguiu mesmo a se estabelecer em 1986, com a publicação do livro “*Parallel distributed processing*” dos autores Rumelhart, Hinton e Williams. Neste livro, eles desenvolveram um algoritmo conhecido como “*Backpropagation*” capaz de atualizar os pesos de uma ou mais camadas, solucionando o problema apresentado por Minsky e Papert.

3.4 - O Neurônio Biológico

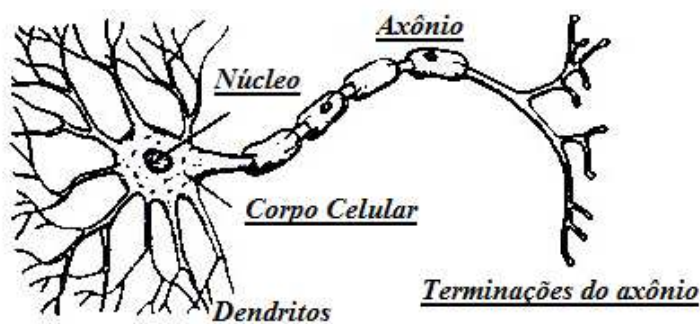
As informações processadas pelo cérebro humano são conduzidas pelos processadores biológicos (neurônios) que operam em paralelo, com a finalidade de ativar as funcionalidades como o pensar e o memorizar (SILVA; SPATTI; FLAUZINO, 2010).

A principal célula do sistema nervoso cerebral é o neurônio biológico que se divide em: corpo celular, dendritos e axônio. A Figura 3.1 ilustra esse neurônio e seus componentes.

Cada um dos componentes do neurônio biológico mostrado na Figura 3.1 possui uma função específica, porém eles se complementam. A função dos dendritos é captar os impulsos nervosos oriundos de outros neurônios e guiá-los até o corpo celular. Este corpo processa a informação originada pelos dendritos

e é responsável por gerar um potencial de ativação que poderá ser ou não enviado para o axônio (SILVA; SPATTI; FLAUZINO, 2010).

Figura 3.1 - Neurônio biológico e seus componentes. Fonte: (ALVES, 2007)



A missão do axônio é propagar os impulsos elétricos para outros neurônios. A união do axônio de um neurônio e o dendrito de outro, recebe o nome de sinapse. E é pelas sinapses que ocorre a união dos neurônios formando a rede neural biológica.

Esse processo é responsável pela maioria das funções do cérebro humano. A capacidade de realizar funções complexas é realizada pela operação de todos os neurônios do nosso sistema nervoso (BRAGA; CARVALHO; LUDEMIR, 2007)

3.5 - Modelo do neurônio artificial

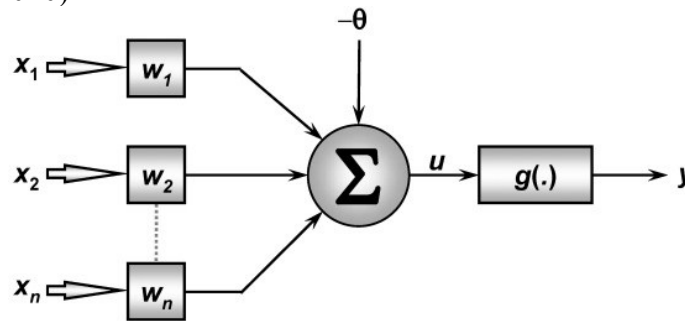
O modelo do neurônio artificial originado do trabalho de McCulloch e Pitts em 1943 foi idealizado em relação ao que se sabia na época sobre o neurônio biológico. Esse modelo possui n terminais de entrada (dendritos) que recebem os valores $\{x_1, x_2, \dots, x_n\}$ e um terminal de saída y que representa o axônio.

As ligações dessas entradas ao corpo celular artificial são realizadas por elementos chamados de pesos $\{w_1, w_2, \dots, w_n\}$ que representam o comportamento das sinapses. Assim, a primeira operação matemática do neurônio artificial é a multiplicação das entradas $\{x_i\}$ dos neurônios pelos seus respectivos pesos sinápticos $\{w_i\}$ (SILVA; SPATTI; FLAUZINO, 2010) e (BRAGA; CARVALHO; LUDEMIR, 2007).

A segunda operação matemática é a determinação da saída do corpo celular denotado por u (potencial de ativação). Essa operação é obtida pela soma ponderada do produto entre as entradas e os pesos sinápticos, subtraída do limiar de ativação (θ). Finalmente, a terceira e a última operação é a determinação da saída y que pode ser obtida pela aplicação de uma função de ativação em u . O objetivo dessa última operação é limitar o valor da amplitude da saída.

A Figura 3.2 mostra o modelo de um neurônio artificial.

Figura 3.2 - Neurônio artificial. Fonte: (SILVA; SPATTI; FLAUZINO, 2010)



Matematicamente, o modelo utilizado por McCulloch e Pitts pode ser obtido pelas Equações (3.1) e (3.2), respectivamente (SILVA; SPATTI; FLAUZINO, 2010).

$$u = \sum_{i=1}^n x_i \cdot w_i - \theta \quad (3.1)$$

$$y = g(u) \quad (3.2)$$

onde:

$g(u)$ – função de ativação aplicada em u .

3.5.1 - Funções de ativação

O objetivo principal das funções de ativação utilizadas em redes neurais artificiais é limitar a saída y dentro de um intervalo de $[0, 1]$, ou, no intervalo de $[-1, 1]$. Essas funções dependem do tipo de rede utilizada e podem ser divididas em funções parcialmente diferenciáveis e funções totalmente diferenciáveis.

As funções degrau (*heavside*), função sinal (*symmetric hard limiter* ou degrau bipolar) e a função rampa são funções parcialmente diferenciáveis.

A função degrau é igual a “0” se o resultado do potencial de ativação é menor do que “0” e se o valor deste potencial é maior ou igual a “0”, o valor dessa função é igual a “1”. A função degrau é obtida pela Equação (3.3) (SILVA; SPATTI; FLAUZINO, 2010). A Figura 3.3 mostra essa função.

$$g(u) = \begin{cases} 1, & \text{se } u \geq 0 \\ 0, & \text{se } u < 0 \end{cases} \quad (3.3)$$

Segundo Silva, Spatti e Flauzino (2010) a função sinal pode ser encontrada pela Equação (3.4). O valor dessa função é igual a “1” se o potencial de ativação é maior do que “0”, “0” se esse potencial é igual a “0” e “- 1” se o potencial de ativação assume valores menores do que “0”. A Figura 3.4 ilustra essa função.

$$g(u) = \begin{cases} 1, & \text{se } u > 0 \\ 0, & \text{se } u = 0 \\ -1, & \text{se } u < 0 \end{cases} \quad (3.4)$$

Figura 3.3 - Função degrau. Fonte: Adaptado (SILVA; SPATTI; FLAUZINO, 2010)

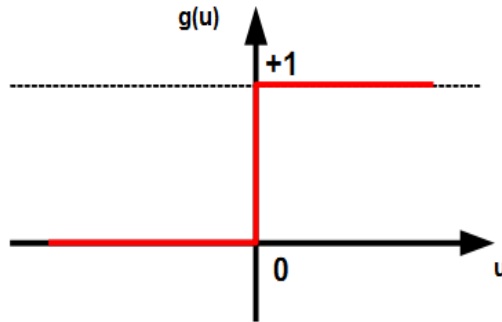
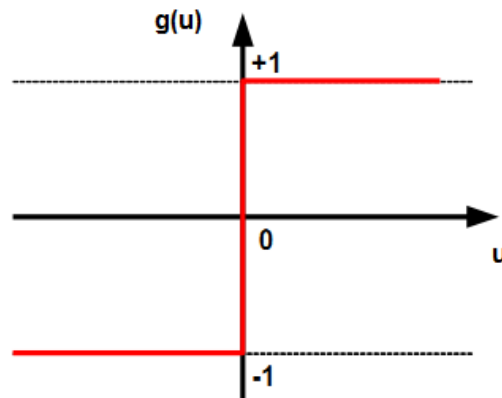


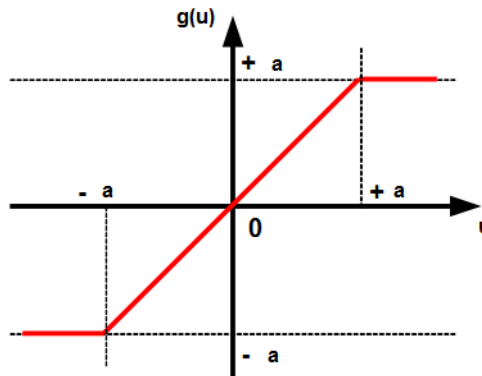
Figura 3.4 - Função sinal. Fonte: Adaptado (SILVA; SPATTI; FLAUZINO, 2010)



A função rampa recebe em sua saída os próprios potenciais de ativação quando esses estão no intervalo de $[-a, a]$, fora desse intervalo a saída recebe “a” se o potencial de ativação é maior do que “a” e “- a” se o potencial de ativação é menor do que “a”. A Equação (3.5) representa essa função (SILVA; SPATTI; FLAUZINO, 2010). A Figura 3.5 mostra a função rampa.

$$g(u) = \begin{cases} a, & \text{se } u > a \\ u, & \text{se } -a \leq u \leq a \\ -a, & \text{se } u < -a \end{cases} \quad (3.5)$$

Figura 3.5 - Função rampa. Fonte: Adaptado (SILVA; SPATTI; FLAUZINO, 2010)



As funções de ativação totalmente diferenciáveis cuja derivada de primeira ordem é conhecida em todos os pontos do seu domínio são divididas em: função logística e função tangente hiperbólica.

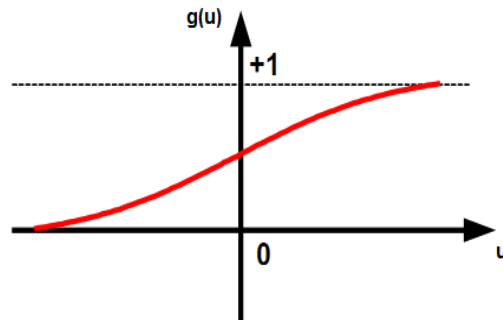
Como mostrado na Figura 3.6, Haykin (2007) citou que o gráfico da função logística possui o formato de um “S”. Essa função é estritamente crescente e assume valores reais entre zero e um. A Equação (3.6) define matematicamente essa função.

$$g(u) = \frac{1}{1 + e^{-\beta u}} \quad (3.6)$$

onde:

β – constante real que define o nível de inclinação da função rampa em relação ao ponto de inflexão.

Figura 3.6 - Função logística. Fonte: Adaptado (SILVA; SPATTI; FLAUZINO, 2010)

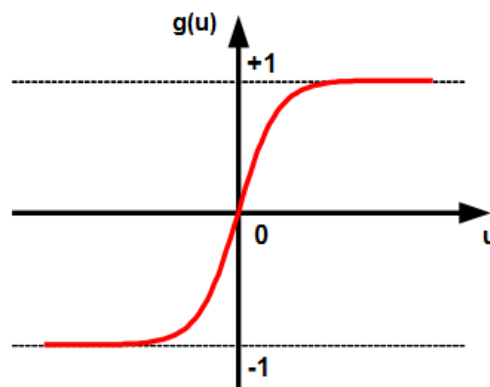


A função tangente hiperbólica assume valores reais no domínio de -1 a 1 . A Equação (3.7) mostra a expressão matemática dessa função.

$$g(u) = \frac{1 - e^{-\beta u}}{1 + e^{-\beta u}} \quad (3.7)$$

Na função tangente hiperbólica, o β também define o nível de inclinação em relação ao ponto de inflexão. A Figura 3.7 ilustra essa função.

Figura 3.7 - Função tangente hiperbólica. Fonte: Adaptado (SILVA; SPATTI; FLAUZINO, 2010)



3.6 - Principais arquiteturas das redes neurais artificiais

Braga, Carvalho e Ludermir (2007) afirmaram que independente da função de ativação escolhida, os neurônios individuais possuem capacidade computacional limitada. Assim, um conjunto de neurônios arranjados em uma rede formam uma arquitetura de RNA's. Esse rearranjo é capaz de resolver problemas de maior complexidade e solucionar a capacidade limitada do neurônio individual.

Para definir um tipo de rede neural artificial é necessário definir o número de camadas neurais, os tipos de conexões entre os neurônios e o algoritmo de aprendizagem usado no treinamento. Essa rede pode ser dividida em camadas. Essas camadas são:

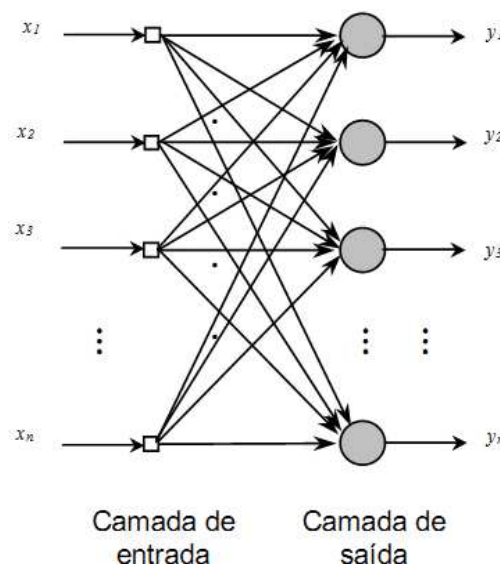
Camada de entrada: responsável em receber os dados oriundos do meio externo que devem ser normalizados para uma melhor precisão das operações matemáticas utilizadas pela rede neural artificial;

Camada escondida: essa camada também chamada de intermediária e/ou oculta é responsável pela extração das informações para depois generalizar; e

Camada de saída: responsável pela saída da rede. Esse resultado é originado do processamento realizado pelos neurônios das camadas anteriores.

A estrutura mais simples é a arquitetura de redes alimentadas adiante com camada única (*feedforward*). Ela é composta de n entradas e m saídas. A Figura 3.8 mostra esse tipo de rede.

Figura 3.8 - Rede *feedforward*. Fonte: Adaptado de Haykin (2007)



Na rede *feedforward* mostrada na Figura 3.8, a informação caminha em um único sentido, ou seja, da entrada para a saída. Essa rede possui uma camada de entrada e uma única camada de neurônios que é a própria camada de saída. Ela é utilizada em reconhecimento de padrões e memórias associativas. A rede *feedforward* é usada nos modelos Perceptron e Adaline. Os algoritmos de aprendizagem utilizados no treinamento desses modelos são as regras de Hebb e Delta, respectivamente (SILVA; SPATTI; FLAUZINO, 2010).

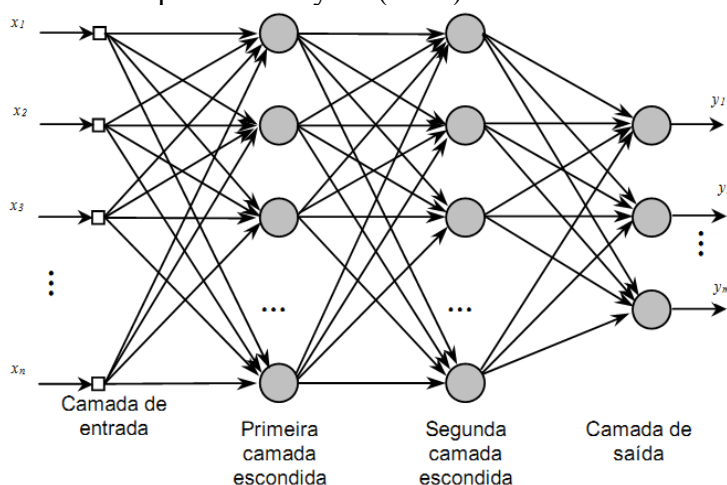
A arquitetura de redes alimentadas diretamente com camadas múltiplas (*feedforward* de camada múltiplas) possui a camada de entrada, uma ou mais camadas ocultas e a camada de saída. A Figura 3.9 ilustra um exemplo desse tipo de rede.

As diversas aplicações da rede *feedforward* de camadas múltiplas são: aproximação de funções, identificação de sistemas, reconhecimento de padrões, otimização, robótica, controle de processos, etc. (SILVA; SPATTI; FLAUZINO, 2010). As principais redes que utilizam essa arquitetura são: *Multilayer Perceptron* (MLP) e Função Base Radial (RBF). Os algoritmos para

o treinamento dessas redes são baseados nas regras delta generalizada e delta competitiva.

É importante salientar que o número de neurônios utilizados nas camadas escondidas depende da complexidade do problema a ser mapeado pela rede.

Figura 3.9 - Exemplo de uma rede *feedforward* de camadas múltiplas.
Fonte: Adaptado de Haykin (2007)



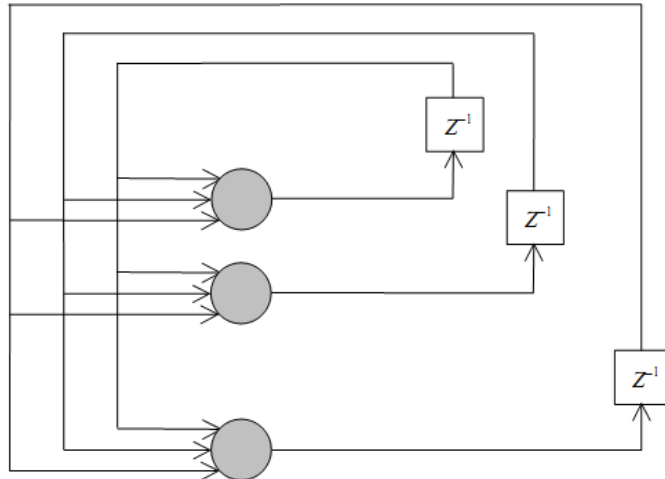
Uma outra arquitetura de redes utilizadas são as redes recorrentes. Elas se diferenciam em relação as redes *feedforward* por terem pelo menos um laço de realimentação (HAYKIN, 2007). Nessa rede as camadas de neurônios da saída são realimentadas dos sinais de entrada para os outros neurônios. A Figura 3.10 mostra um exemplo de rede recorrente.

A rede recorrente pode ser aplicada em previsão de séries temporais, otimização, identificação de sistemas, controle de processos, dentre outras aplicações. Nessa rede os algoritmos de aprendizado usados no treinamento são minimização de funções de energia e a regra delta generalizada, respectivamente (SILVA; SPATTI; FLAUZINO, 2010).

Na Figura 3.10, o atraso unitário é representado por z^{-1} . Assim, incluindo os atrasos pela realimentação, as saídas atuais são obtidas levando em consideração os valores anteriores da saída. Como citado por Haykin (2007), o

resultado dessa operação melhora a capacidade de aprendizagem da rede. A rede de Hopfield é um exemplo de rede recorrente.

Figura 3.10 - Exemplo de uma rede recorrente. Fonte: Adaptado de Haykin (2007)



3.7 - Tipos de treinamento das redes neurais artificiais

As RNAs apresentam uma característica muito importante, que é a capacidade de aprender utilizando exemplos e generalizar essa informação. Segundo Silva, Spatti e Flauzino (2010), essa característica é alcançada pelo ajuste dos pesos sinápticos e dos limiares, de forma que, a aplicação de um conjunto de entradas produza um conjunto de saídas desejadas.

A forma de como esses pesos e os limiares são ajustados, define o tipo de algoritmo utilizado no treinamento. Esse treinamento pode ser do tipo supervisionado ou não supervisionado.

3.7.1 - Treinamento supervisionado

O treinamento do tipo supervisionado é realizado sob a supervisão de um “professor” externo que tem a função de monitorar para cada entrada a resposta da rede, conhecendo previamente a saída desejada.

Braga, Carvalho e Ludermir (2007) mencionaram que os algoritmos usados para esse treinamento são a regra delta e a sua generalização para as redes de múltiplas camadas e o algoritmo *backpropagation*.

3.7.2 - Treinamento não supervisionado

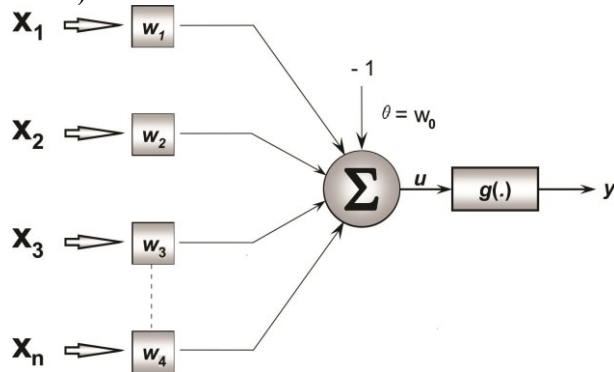
No treinamento não supervisionado não existe necessidade da saída desejada em relação aos estímulos da entrada. Assim, a rede se auto organiza em relação as particularidades do conjunto de entrada (SILVA; SPATTI; FLAUZINO, 2010).

Esse tipo de treinamento utiliza a regra de aprendizagem competitiva (HAYKIN, 2007).

3.8 - Perceptron camada única

O Perceptron proposto por Rosenblatt em 1958 é a configuração de rede neural mais simples usada na classificação de padrões linearmente separáveis. Essa simplicidade é justificada utilizando uma única camada neural com apenas um neurônio artificial. A Figura 3.11 ilustra essa rede com n entradas e somente uma saída.

Figura 3.11 - Perceptron. Fonte: (SILVA; SPATTI; FLAUZINO, 2010)



A operação da rede Perceptron é baseada no modelo neural de McCulloch e Pitts. Em termos matemáticos, o processamento em camada única pode ser obtido pelas Equações (3.8) e (3.9), respectivamente (SILVA; SPATTI; FLAUZINO, 2010).

$$u = \sum_{i=1}^n w_i \cdot x_i - \theta \quad (3.8)$$

$$y = g(u) \quad (3.9)$$

onde:

x_i – entradas da rede, cujo os valores oriundos das medidas externas podem ser reais ou binários;

w_i – peso (ponderação) associado a i -ésima entrada, ele é um parâmetro real iniciado aleatoriamente;

θ – limiar de ativação iniciado aleatoriamente, cujo valor é real;

$g(\cdot)$ – função de ativação que pode ser degrau ou degrau bipolar; e

y – saída que assume um valor binário.

Considerando que a saída da rede Perceptron só pode assumir dois valores é possível verificar que essa rede, pode ser utilizada como um discriminador linear para separar duas classes linearmente separáveis. Matematicamente, a saída da rede Perceptron adotando a função de ativação degrau bipolar pode ser encontrada pela Equação (3.10) (SILVA; SPATTI; FLAUZINO, 2010).

$$y = \begin{cases} 1, & \text{se } \sum w_i \cdot x_i - \theta \geq 0 \\ -1, & \text{se } \sum w_i \cdot x_i - \theta < 0 \end{cases} \quad (3.10)$$

O ajuste dos pesos e o limiar do Perceptron camada única é realizado por um treinamento supervisionado. Mais especificamente, usando a regra de aprendizado de Hebb de 1949 (KOVÁCS, 1996).

Esse ajuste é realizado da seguinte forma: se o resultado da saída do Perceptron (y) é igual ao valor da saída desejada atribuída a k -ésima amostra, os pesos e os limiares não sofrem alterações. Caso (y) seja diferente de ($d^{(k)}$) para a k -ésima amostra, esses parâmetros são ajustados pelas Equações (3.11) e (3.12), respectivamente (SILVA; SPATTI; FLAUZINO, 2010).

$$w_i^{atual} = w_i^{anterior} + \eta \cdot (d^{(k)} - y) \cdot \mathbf{x}^{(k)} \quad (3.11)$$

$$\theta_i^{atual} = \theta_i^{anterior} + \eta \cdot (d^{(k)} - y) \cdot \mathbf{x}^{(k)} \quad (3.12)$$

Nesse processo, os pesos e os limiares são alterados até que a saída produzida pela rede Perceptron é similar a saída desejada de cada amostra.

Para facilitar a implementação computacional, as expressões usadas nas Equações (3.11) e (3.12) podem ser representadas por uma única expressão vetorial como mostra a Equação (3.13) (SILVA; SPATTI; FLAUZINO, 2010).

$$\mathbf{w}^{atual} = \mathbf{w}^{anterior} + \eta \cdot (d^{(k)} - y) \cdot \mathbf{x}^{(k)} \quad (3.13)$$

onde:

$\mathbf{w} = [\theta \ w_1 \ w_2 \ \dots \ w_n]^T$ – vetor contendo o limiar e os pesos;

$\mathbf{x}(k) = [-1 \ x_1^{(k)} \ x_2^{(k)} \ \dots \ x_n^{(k)}]^T$ – k -ésima amostra de treinamento;

$d^{(k)}$ – valor desejado para a k -ésima amostra de treinamento;

y – valor da saída produzida pela rede Perceptron; e

η – constante que define a taxa de aprendizagem que varia de 0 a 1.

Para Braga, Carvalho e Ludermir (2007), a taxa de aprendizagem η define a velocidade com que o vetor de pesos é modificado. Assim, para cada problema é necessário ter atenção em relação ao aumento dessa taxa, pois a consequência pode ser a instabilidade da rede.

3.9 - Método do gradiente descendente (regra delta)

O método do gradiente descendente também conhecido como regra delta foi desenvolvido por Widrow & Hoff em 1960. O objetivo desse método é minimizar o erro entre a saída desejada (d) e a saída do combinador linear (potencial de ativação) (u) da rede com a finalidade de ajustar o vetor de pesos $\mathbf{w} = [\theta \ w_1 \ w_2 \ \dots \ w_n]^T$ da rede. Em suma, o objetivo é encontrar um \mathbf{w}^* ótimo de tal forma que o erro quadrático $\{E(\mathbf{w}^*)\}$ é o mínimo possível sobre todas as amostras.

Considerando p amostras, a função erro quadrático é encontrada pela Equação (3.14).

$$E(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^p \left(d^{(k)} - u \right)^2 \quad (3.14)$$

Substituindo a Equação (3.8) na Equação (3.14) obtém-se as Equações (3.15) e (3.16), respectivamente.

$$E(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^p \left(d^{(k)} - \left(\sum_{i=1}^n w_i \cdot x_i^{(k)} - \theta \right) \right)^2 \quad (3.15)$$

$$E(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^p \left(d^{(k)} - (\mathbf{w}^T \cdot \mathbf{x}^{(k)} - \theta) \right)^2 \quad (3.16)$$

Para encontrar o valor ótimo para o erro quadrático é aplicado na Equação (3.16) o operador gradiente em relação ao vetor \mathbf{w} . A Equação (3.17) mostra essa operação.

$$\nabla E(\mathbf{w}) = \frac{\partial E(\mathbf{w})}{\partial \mathbf{w}} \quad (3.17)$$

Aplicando a Equação (3.16) na Equação (3.17) obtém-se a Equação (3.18)

$$\nabla E(\mathbf{w}) = \sum_{k=1}^p \left(d^{(k)} - (\mathbf{w}^T \cdot \mathbf{x}^{(k)} - \theta) \right) \cdot (-\mathbf{x}^{(k)}) \quad (3.18)$$

Substituindo-se a Equação (3.8) na Equação (3.18) obtém-se a Equação (3.19).

$$\nabla E(\mathbf{w}) = \sum_{k=1}^p \left(d^{(k)} - u \right) \cdot (-\mathbf{x}^{(k)}) \quad (3.19)$$

Finalmente, o ajuste dos pesos pode ser realizado em direção oposta ao gradiente descendente com o objetivo de minimizar o erro médio quadrático. Logo, a variação $\Delta \mathbf{w}$ a ser acrescida na correção dos pesos é obtida pela Equação (3.20).

$$\Delta \mathbf{w} = -\eta \cdot \nabla E(\mathbf{w}) \quad (3.20)$$

Substituindo-se a Equação (3.19) na Equação (3.20), obtém-se a variação $\Delta \mathbf{w}$ na Equação (3.21).

$$\Delta \mathbf{w} = \eta \cdot \sum_{k=1}^p (d^{(k)} - u) \cdot \mathbf{x}^{(k)} \quad (3.21)$$

Portanto, a Equação (3.22) representa a atualização de \mathbf{w} após a apresentação de cada k -ésimo padrão de treinamento.

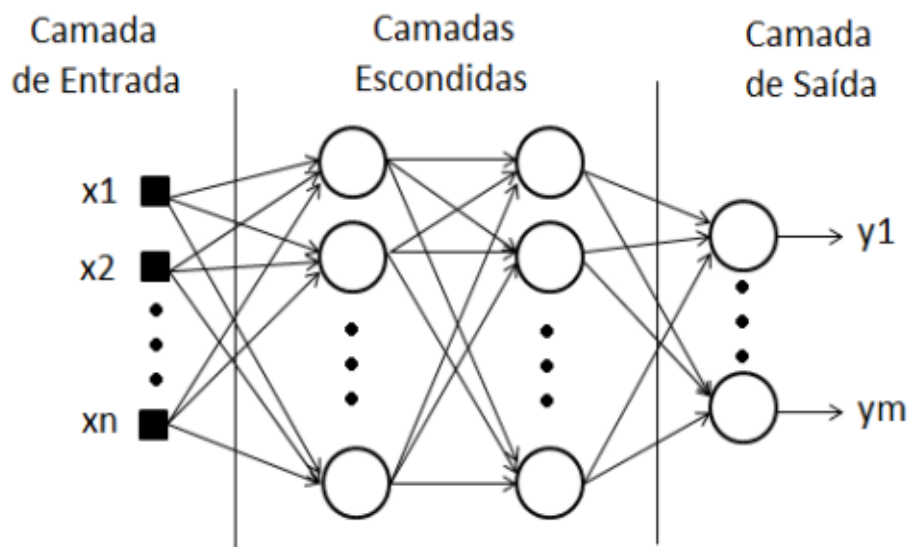
$$\mathbf{w}^{atual} = \mathbf{w}^{anterior} + \eta \cdot \sum_{k=1}^p (d^{(k)} - u) \cdot \mathbf{x}^{(k)} \quad (3.22)$$

3.10 - Redes Perceptron múltiplas camadas

A rede Perceptron múltiplas camadas (PMC) do inglês *Multilayer Perceptron* (MLP) consiste em uma camada de entrada, uma ou mais camadas intermediárias e uma camada de saída. Os neurônios que pertencem a camada de entrada são denominados unidades de entrada. Esses neurônios propagam os valores das entradas para as camadas seguintes sem modificação. As camadas intermediárias, escondidas ou ocultas transmitem as informações por meio das conexões entre as camadas de entrada e saída. Finalmente, a camada de saída

transmite a resposta da rede neural à entrada aplicada na camada de entrada (HAYKIN, 2007). A Figura 3.12 mostra a arquitetura de uma rede MLP com duas camadas intermediárias.

Figura 3.12 - Exemplo de uma rede MLP com duas camadas ocultas. Fonte: Cassimiro (2013)



Na rede Perceptron múltiplas camadas, os neurônios são conectados por arestas e cada aresta está associada a um peso. A comunicação entre os neurônios é realizada por meio dessas interconexões ponderadas. Porém, a comunicação ocorre somente entre as unidades de camadas distintas.

Cada neurônio da MLP possui uma função de ativação. Esta função pode ser linear, logística ou tangente hiperbólica. A utilização dessas funções na rede Perceptron múltiplas camadas é porque elas são totalmente diferenciáveis.

Uma das características relacionadas as redes MLP é que elas apresentam variações em relação ao número de camadas, neurônios da camada intermediária, tipo de conexão determinado pelas sinapses e a função de ativação.

A rede *Multilayer Perceptron* utiliza um treinamento do tipo supervisionado para indicar a resposta desejada a um padrão de entrada apresentado a ela durante a fase de aprendizagem. A resposta desejada e aquelas

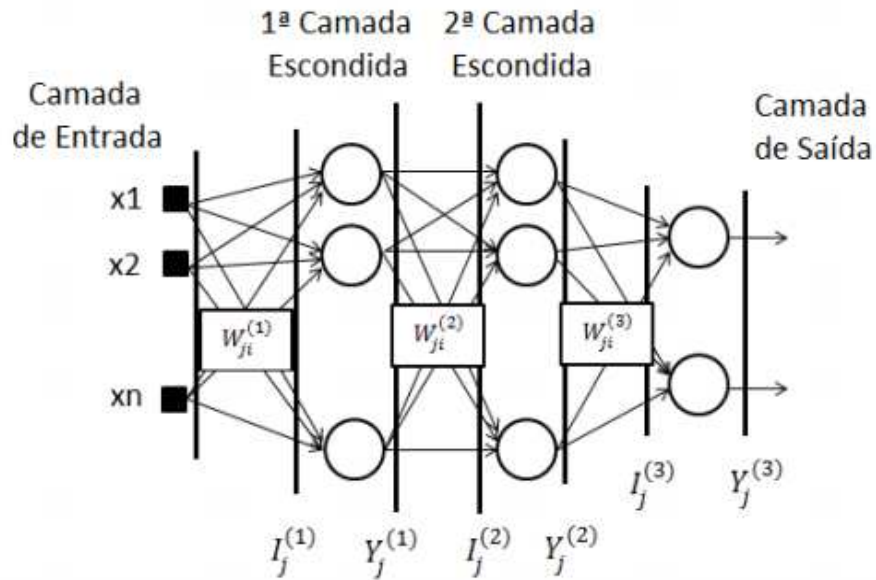
produzidas pelos neurônios de saída são calculadas e subsequentemente utilizadas para ajustar os pesos e os limiares de todos os seus neurônios. O método de aprendizagem utilizado para esse ajuste é o algoritmo *backpropagation* ou o algoritmo de retro-propagação do erro.

3.10.1 - Algoritmo *backpropagation*

Para o treinamento da rede MLP, o método de aprendizagem é o algoritmo *backpropagation*. As fases deste algoritmo são divididas em dois passos: um passo para frente a propagação (*forward*) e um passo para trás, a retro-propagação (*backward*). Na primeira fase conhecida como *forward*, as entradas são apresentadas e propagadas por meio da rede, camada por camada, calculando a saída de cada neurônio. Durante essa fase, os pesos são fixos e a saída calculada é comparada com a saída desejada, resultando em um erro para cada unidade. Na fase *backward*, o erro calculado é propagado da camada de saída para a camada de entrada e os pesos são ajustados de acordo com a regra de correção de erro.

Para uma melhor compreensão do algoritmo *backpropagation* existe a necessidade de definir diversas variáveis auxiliares que serão usadas para o desenvolvimento matemático. A Figura 3.13 ilustra o conjunto de variáveis utilizadas para esse entendimento (SILVA; SPATTI; FLAUZINO, 2010).

Figura 3.13 - Notação para a derivação do algoritmo *backpropagation*. Fonte: Cassimiro (2013).



$W_{ji}^{(L)}$ – peso sináptico do j -ésimo neurônio da camada (L) ao i -ésimo neurônio da camada ($L - 1$);

$I_j^{(L)}$ – potencial de ativação do j -ésimo neurônio da camada (L);

$\theta_j^{(L)}$ – limiar de ativação do j -ésimo neurônio da camada (L); e

$Y_j^{(L)}$ – saída do j -ésimo neurônio da camada (L).

Considerando a Figura 3.13, o desenvolvimento matemático para o potencial de ativação da rede MLP é obtido pelas Equações (3.23), (3.24) e (3.25), respectivamente (SILVA; SPATTI; FLAUZINO, 2010).

$$I_j^{(1)} = \sum_{i=0}^n W_{ji}^{(1)} \cdot x_i \quad (3.23)$$

$$I_j^{(2)} = \sum_{i=0}^{n_1} W_{ji}^{(2)} \cdot Y_i^{(1)} \quad (3.24)$$

onde:

n_1 – número de neurônios da primeira camada escondida.

$$I_j^{(3)} = \sum_{i=0}^{n_2} W_{ji}^{(3)} \cdot Y_i^{(2)} \quad (3.25)$$

onde:

n_2 – número de neurônios da segunda camada escondida.

De forma análoga, o cálculo para as saídas é encontrado pelas Equações (3.26), (3.27) e (3.28), respectivamente:

$$Y_j^{(1)} = g(I_j^{(1)}) \quad (3.26)$$

$$Y_j^{(2)} = g(I_j^{(2)}) \quad (3.27)$$

$$Y_j^{(3)} = g(I_j^{(3)}) \quad (3.28)$$

onde:

$g(.)$ – função de ativação.

Segundo Silva, Spatti e Flauzino (2010), após calcular as saídas, o próximo passo é definir o erro entre a saída desejada e a saída da rede. Assim, considerando a k -ésima amostra de treinamento para a topologia ilustrada na Figura 3.13, o erro quadrático é definido pela Equação (3.29).

$$E(k) = \frac{1}{2} \sum_{j=1}^N (d_j(k) - Y_j^{(3)}(k))^2 \quad (3.29)$$

onde:

$Y_j^{(3)}(k)$ – valor produzido pelo j -ésimo neurônio da saída da rede considerando a k -ésima amostra de treinamento; e

$d_j(k)$ – k -ésimo valor desejado.

Como mostrado na Equação (3.30), a medida do desempenho global da rede para um conjunto de treinamento composto por p amostras pode ser calculada pelo erro quadrático médio, (SILVA; SPATTI; FLAUZINO, 2010).

$$EM = \frac{1}{p} \sum_{k=1}^p E(k) \quad (3.30)$$

Para verificar como a atualização dos pesos é realizada no algoritmo *backpropagation*, ela é dividida em etapas: o ajuste dos pesos das camadas de saída e intermediária.

a) Ajuste dos pesos sinápticos da camada de saída

A finalidade do ajuste da matriz de pesos $W_{ji}^{(3)}$ da camada de saída é minimizar o erro entre a saída da rede em relação a saída desejada.

Aplicando-se na Equação (3.29) a definição do gradiente e a regra de diferenciação em cadeia, obtém-se a Equação (3.31) (SILVA; SPATTI; FLAUZINO, 2010).

$$\nabla E^{(3)} = \frac{\partial E}{\partial W_{ji}^{(3)}} = \frac{\partial E}{\partial Y_j^{(3)}} \cdot \frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} \cdot \frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}} \quad (3.31)$$

O termo $\frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}}$ da Equação (3.31) pode ser obtido a partir da Equação (3.23). O resultado dessa operação é mostrado na Equação (3.32).

$$\frac{\partial I_j^{(3)}}{\partial W_{ji}^{(3)}} = Y_i^{(2)} \quad (3.32)$$

De forma análoga, os termos $\frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}}$ e $\frac{\partial E}{\partial W_{ji}^{(3)}}$ podem ser encontrados pelas Equações (3.28) e (3.29), respectivamente. O resultado dessas operações pode ser obtido pelas Equações (3.33) e (3.34), respectivamente.

$$\frac{\partial Y_j^{(3)}}{\partial I_j^{(3)}} = g'(I_j^{(3)}) \quad (3.33)$$

$$\frac{\partial E}{\partial Y_j^{(3)}} = -(d_j - Y_j^{(3)}) \quad (3.34)$$

onde:

$g'(\cdot)$ – derivada de primeira ordem da função de ativação.

Substituindo as Equações de (3.32) a (3.34) na Equação (3.31) obtém-se a Equação (3.35).

$$\frac{\partial E}{\partial W_{ji}^{(3)}} = -(d_j - Y_j^{(3)}) \cdot g'(I_j^{(3)}) \cdot Y_i^{(2)} \quad (3.35)$$

Assim, com a finalidade de minimizar o erro, o ajuste da matriz de pesos $W_{ji}^{(3)}$ da camada de saída pode ser obtido em relação a direção oposta do gradiente. A Equação (3.36) mostra a forma de como é encontrado o incremento da matriz de pesos da camada de saída.

$$\Delta W_{ji}^{(3)} = -\eta \cdot \frac{\partial E}{\partial W_{ji}^{(3)}} = \eta \cdot \delta_j^{(3)} \cdot Y_i^{(2)} \quad (3.36)$$

onde:

$\delta_j^{(3)}$ – gradiente local em relação ao j -ésimo neurônio da camada de saída.

O $\delta_j^{(3)}$ é obtido pela Equação (3.37).

$$\delta_j^{(3)} = (d_j - Y_j^{(3)}) \cdot g'(I_j^{(3)}) \quad (3.37)$$

Finalmente, a Equação (3.38) mostra como os pesos da camada de saída podem ser ajustados.

$$W_{ji}^{(3)}(t+1) = W_{ji}^{(3)}(t) + \eta \cdot \delta_j^{(3)} \cdot Y_i^{(2)} \quad (3.38)$$

b) Ajuste dos pesos sinápticos das camadas intermediárias

Os neurônios das camadas intermediárias não possuem acesso aos valores desejados para as suas saídas. Assim, o sinal de erro para um neurônio da camada intermediária é determinado em termos dos sinais de erro de todos os neurônios que o neurônio dessa camada escondida está conectado. Em outras

palavras, os ajustes são baseados nos valores ponderados daqueles pesos sinápticos que já foram ajustados para os neurônios da camada de saída.

No processo de treinamento, a finalidade da segunda camada intermediária ou escondida é o ajuste da matriz de pesos $W_{ji}^{(2)}$. O objetivo deste ajuste é minimizar o erro entre a saída produzida pela rede em relação à retropropagação do erro originado da camada neural de saída. A Equação (3.39) pode ser utilizada para o desenvolvimento do ajuste da matriz de pesos.

$$\nabla E^{(2)} = \frac{\partial E}{\partial W_{ji}^{(2)}} = \frac{\partial E}{\partial Y_j^{(2)}} \cdot \frac{\partial Y_j^{(2)}}{\partial I_j^{(2)}} \cdot \frac{\partial I_j^{(2)}}{\partial W_{ji}^{(2)}} \quad (3.39)$$

As Equações (3.40) e (3.41) podem ser obtidas utilizando as Equações (3.24) e (3.27), respectivamente.

$$\frac{\partial I_j^{(2)}}{\partial W_{ji}^{(2)}} = Y_i^{(1)} \quad (3.40)$$

$$\frac{\partial Y_j^{(2)}}{\partial I_j^{(2)}} = g'(I_j^{(2)}) \quad (3.41)$$

O termo $\frac{\partial E}{\partial Y_j^{(2)}}$ da Equação (3.39) pode ser obtido pela Equação (3.42).

$$\frac{\partial E}{\partial Y_j^{(2)}} = \sum_{k=1}^{n_3} \underbrace{\frac{\partial E}{\partial I_k^{(3)}} \cdot \frac{\partial I_k^{(3)}}{\partial Y_j^{(2)}}}_{\text{parcela(i)}} = \underbrace{\sum_{k=1}^{n_3} \frac{\partial E}{\partial I_k^{(3)}} \cdot \frac{\partial \left(\sum_{k=1}^{n_3} W_{kj}^{(3)} \cdot Y_j^{(2)} \right)}{\partial Y_j^{(2)}}}_{\text{parcela(ii)}} \quad (3.42)$$

Observando a Equação (3.42) verifica-se que o argumento em relação a parcela (ii) é o próprio $W_{kj}^{(3)}$. Portanto, a Equação (3.43) apresenta o resultado dessa operação.

$$\frac{\partial E}{\partial Y_j^{(2)}} = \underbrace{\sum_{k=1}^{n_3} \frac{\partial E}{\partial I_k^{(3)}}}_{\text{parcela (i)}} \cdot \underbrace{W_{kj}^{(3)}}_{\text{parcela (ii)}} \quad (3.43)$$

A parcela (i) da Equação (3.43) é obtida pela multiplicação das Equações (3.33) e (3.34). O resultado dessa multiplicação é o gradiente local $\delta_j^{(3)}$. Substituindo-se esse gradiente na parcela (i) da Equação (3.43) obtém-se a Equação (3.44).

$$\frac{\partial E}{\partial Y_j^{(2)}} = -\sum_{k=1}^{n_3} \delta_k^{(3)} \cdot W_{kj}^{(3)} \quad (3.44)$$

Contudo, substituindo as Equações (3.40), (3.41) e (3.43) na Equação (3.39) obtém-se a Equação (3.45).

$$\frac{\partial E}{\partial W_{ji}^{(2)}} = -\left(\sum_{k=1}^{n_3} \delta_k^{(3)} \cdot W_{kj}^{(3)}\right) \cdot g'(I_j^{(2)}) \cdot Y_i^{(1)} \quad (3.45)$$

Como mostrado na Equação (3.46), o ajuste da matriz $W_{ji}^{(2)}$ pode ser realizado na direção oposta ao gradiente com o objetivo de minimizar o erro.

$$\Delta W_{ji}^{(2)} = -\eta \cdot \frac{\partial E}{\partial W_{ji}^{(2)}} \Leftrightarrow \Delta W_{ji}^{(2)} = \eta \cdot \delta_j^{(2)} \cdot Y_i^{(1)} \quad (3.46)$$

onde:

$\delta_j^{(2)}$ – gradiente local em relação ao j -ésimo neurônio da segunda camada intermediária.

O $\delta_j^{(2)}$ é obtido pela Equação (3.47).

$$\delta_j^{(2)} = \left(\sum_{k=1}^{n_3} \delta_k^{(3)} \cdot W_{kj}^{(3)} \right) \cdot g'(I_j^{(2)}) \quad (3.47)$$

É importante ressaltar neste ponto a essência do método *backpropagation*, pois os pesos $W_{ji}^{(3)}$ foram ajustados baseados em valores passados de erro e estes erros são utilizados para atualizar a segunda camada intermediária, ou seja, leva-se em consideração a retropropagação do erro originado nos neurônios de saída.

Finalmente, a atualização da matriz de pesos da segunda camada intermediária pode ser obtida pela Equação (3.48).

$$W_{ji}^{(2)}(t+1) = W_{ji}^{(2)}(t) + \eta \cdot \delta_j^{(2)} \cdot Y_i^{(1)} \quad (3.48)$$

A finalidade do ajuste dos pesos sinápticos da primeira camada escondida é ajustar a matriz de pesos $W_{ji}^{(1)}$. O objetivo do ajuste dessa matriz é minimizar o erro entre a saída produzida pela rede e a retropropagação do erro originado do ajuste da segunda camada escondida. A Equação (3.49) pode ser utilizada para o desenvolvimento do ajuste da matriz de pesos.

$$\nabla E^{(1)} = \frac{\partial E}{\partial W_{ji}^{(1)}} = \frac{\partial E}{\partial Y_j^{(1)}} \cdot \frac{\partial Y_j^{(1)}}{\partial I_j^{(1)}} \cdot \frac{\partial I_j^{(1)}}{\partial W_{ji}^{(1)}} \quad (3.49)$$

As Equações (3.50) e (3.51) podem ser obtidas utilizando as Equações (3.23) e (3.26), respectivamente.

$$\frac{\partial I_j^{(1)}}{\partial W_{ji}^{(1)}} = X_i \quad (3.50)$$

$$\frac{\partial Y_j^{(1)}}{\partial I_j^{(1)}} = g'(I_j^{(1)}) \quad (3.51)$$

O termo $\frac{\partial E}{\partial Y_j^{(1)}}$ pode ser encontrado usando a Equação (3.52).

$$\frac{\partial E}{\partial Y_j^{(1)}} = \sum_{k=1}^{n_2} \frac{\partial E}{\partial I_k^{(2)}} \cdot \frac{\partial I_k^{(2)}}{\partial Y_j^{(1)}} = \underbrace{\sum_{k=1}^{n_2} \frac{\partial E}{\partial I_k^{(2)}}}_{\text{parcela (i)}} \cdot \underbrace{\frac{\partial \left(\sum_{k=1}^{n_2} W_{kj}^{(2)} \cdot Y_j^{(1)} \right)}{\partial Y_j^{(1)}}}_{\text{parcela (ii)}} \quad (3.52)$$

Aplicando a derivada parcial na parcela (ii) da Equação (3.52), verifica-se que o resultado é o valor de $W_{kj}^{(2)}$. A Equação (3.53) mostra a substituição deste valor.

$$\frac{\partial E}{\partial Y_j^{(1)}} = \underbrace{\sum_{k=1}^{n_2} \frac{\partial E}{\partial I_k^{(2)}}}_{\text{parcela (i)}} \cdot \underbrace{W_{kj}^{(2)}}_{\text{parcela (ii)}} \quad (3.53)$$

Observa-se na Equação (3.53) que a parcela (i) pode ser encontrada pela multiplicação das Equações (3.41) e (3.42). O resultado dessa multiplicação é o valor do gradiente local $\delta_j^{(2)}$. Logo, a Equação (3.53) pode ser reescrita pela Equação (3.54)

$$\frac{\partial E}{\partial Y_j^{(1)}} = -\sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)} \quad (3.54)$$

Contudo, substituindo as Equações (3.50), (3.51) e (3.54) na Equação (3.49) obtém-se a Equação (3.55).

$$\frac{\partial E}{\partial W_{ji}^{(1)}} = -\left(\sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)}\right) \cdot g'(I_j^{(1)}) \cdot X_i \quad (3.55)$$

Assim, como mostrado na Equação (3.56) o ajuste da matriz $W_{ji}^{(1)}$ pode ser realizado na direção oposta ao gradiente com o objetivo de minimizar o erro.

$$\Delta W_{ji}^{(1)} = -\eta \cdot \frac{\partial E}{\partial W_{ji}^{(1)}} \Leftrightarrow \Delta W_{ji}^{(1)} = \eta \cdot \delta_j^{(1)} \cdot X_i \quad (3.56)$$

onde:

$\delta_j^{(1)}$ – gradiente local em relação ao j -ésimo neurônio da primeira camada intermediária.

O $\delta_j^{(1)}$ é obtido pela Equação (3.57).

$$\delta_j^{(1)} = \left(\sum_{k=1}^{n_2} \delta_k^{(2)} \cdot W_{kj}^{(2)} \right) \cdot g'(I_j^{(1)}) \quad (3.57)$$

Finalmente, a atualização da matriz de pesos da primeira camada intermediária pode ser obtida pela Equação (3.58).

$$W_{ji}^{(1)}(t+1) = W_{ji}^{(1)}(t) + \eta \cdot \delta_j^{(1)} \cdot X_i \quad (3.58)$$

É importante ressaltar que os pesos só são ajustados na fase de treinamento em que são aplicados os passos *forward* e *backward*. Na fase de teste nenhum tipo de ajuste é efetuado na rede, somente é processada a fase *forward* com o objetivo de gerar as suas saídas.

Diversas variações do algoritmo *backpropagation* têm sido propostas com a finalidade de tornar o processo de convergência da rede mais eficiente. Uma dessas variações é a inserção do termo momentum.

O objetivo do uso do termo momentum é acelerar a convergência da rede. Formalmente, a atualização da matriz de pesos com esse termo, considerando à L -ésima camada é encontrada pela Equação (3.59) (SILVA; SPATTI; FLAUZINO, 2010).

$$W_{ji}^{(L)}(t+1) = W_{ji}^{(L)}(t) + \underbrace{\alpha \left(W_{ji}^{(L)}(t) - W_{ji}^{(L)}(t-1) \right)}_{\text{termo de momentum}} + \underbrace{\eta \cdot \delta_j^{(L)} \cdot Y_i^{(L-1)}}_{\text{taxa de aprendizagem}} \quad (3.59)$$

onde:

α – taxa de momentum.

O valor de α varia entre zero e um.

Outras formas de melhorar a eficiência ou o desempenho de uma rede MLP é utilizar a validação cruzada e a normalização dos dados.

A validação cruzada utiliza uma metodologia para testar o poder de generalização da rede que particiona o conjunto de dados em conjunto de teste e treinamento. Essa metodologia deve ser seguida da seguinte maneira:

I – Particionar o conjunto em dois subconjuntos:

- O conjunto de treinamento utilizado no treinamento da rede deve possuir de 80 a 90% do total dos dados; e
- Para verificar se a rede está generalizando de forma satisfatória, o conjunto de teste deve possuir de 10 a 20% dos dados.

II – Para ajustar todas as topologias candidatas, utilizar o mesmo conjunto de treinamento; e

III – Após o treinamento utilizar o conjunto de teste para avaliar e escolher (validar) o melhor modelo;

- Em certos casos, uma determinada topologia pode ter desempenho satisfatório no treinamento, mas durante a fase de teste a rede pode produzir resultados não satisfatórios (*overfitting*).
- O aumento dos neurônios e das camadas não significa que a rede irá generalizar melhor; e
- Para duas topologias que generalizam com o mesmo grau de precisão deve-se optar por aquela com o menor número de neurônios.

Na normalização dos dados, deve-se levar em consideração o tipo de função de ativação utilizada. Por exemplo, se é usada a função logística, os dados de entrada da rede devem ser normalizados de 0 a 1. Ao utilizar a função tangente hiperbólica, a normalização deve ser de -1 a 1 . Em outras palavras, essa normalização é realizada dentro das faixas de variações dinâmicas das funções de ativação.

Em termos computacionais, esses procedimentos podem melhorar a eficiência da rede MLP.

3.11 - Considerações finais deste capítulo

Este capítulo iniciou com uma rápida introdução às redes neurais artificiais. Em seguida foi apresentado um breve histórico sobre essas redes, o neurônio biológico, o modelo do neurônio artificial, as principais arquiteturas das redes neurais artificiais e os tipos de treinamento dessas redes. Depois, foi descrito o Perceptron camada única e o método do gradiente descendente (regra delta). E finalmente, foi apresentada a rede Perceptron de múltiplas camadas e o algoritmo de treinamento *backpropagation* que será utilizado nesse trabalho.

O próximo capítulo descreve o método proposto. Finalmente, são realizadas as considerações finais do capítulo.

CAPÍTULO 4

4. MÉTODO PROPOSTO

4.1 - Introdução

Para atender a necessidade de um sistema de reconhecimento de edifícios capaz de ser imune a diferentes transformações geométricas e fotométricas das imagens, é apresentado nesse capítulo o método proposto. Este método consiste de 4 etapas. A primeira realiza o pré-processamento da imagem colorida. Com a finalidade de extrair as características da imagem pré-processada, na segunda etapa aplica-se nessa imagem a representação de wavelet Gabor. O resultado da imagem filtrada usando essa representação é transformado em um vetor de características. Na terceira etapa utiliza-se a transformada wavelet para reduzir a dimensionalidade deste vetor. E finalmente, na quarta etapa são realizados o treinamento e os testes usando a MLP. A Figura 4.1 ilustra o método proposto nessa tese.

Esse capítulo descreve o método proposto nesse trabalho. Finalmente, são realizadas as considerações finais deste capítulo.

Figura 4.1 - Diagrama em blocos do sistema de reconhecimento de edifícios proposto.



4.2 - Pré-Processamento

Nessa etapa é realizada a leitura da imagem no banco de dados de edifícios. Após essa leitura, a imagem é convertida para 256 níveis de cinza. A Figura 4.2 mostra a imagem original e o resultado obtido nessa etapa.

Figura 4.1- (a) imagem original; e (b) imagem em níveis de cinza.



(a)



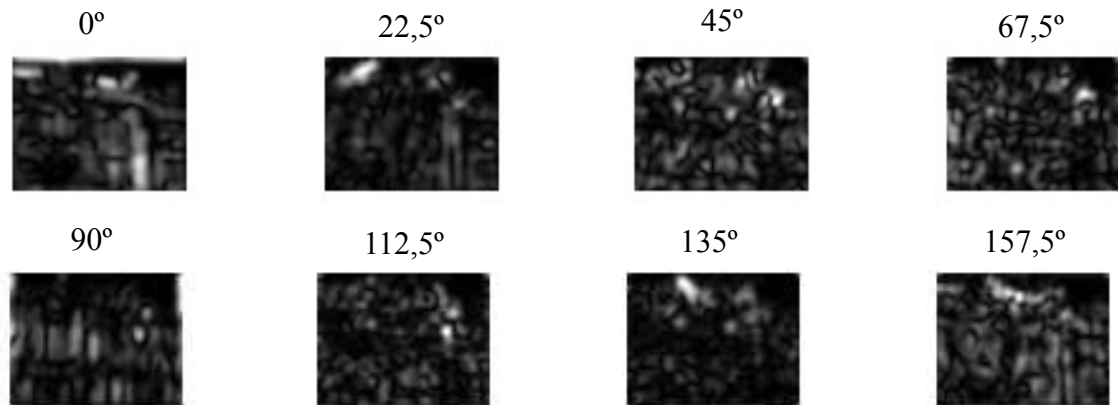
(b)

4.3 - Extração das características

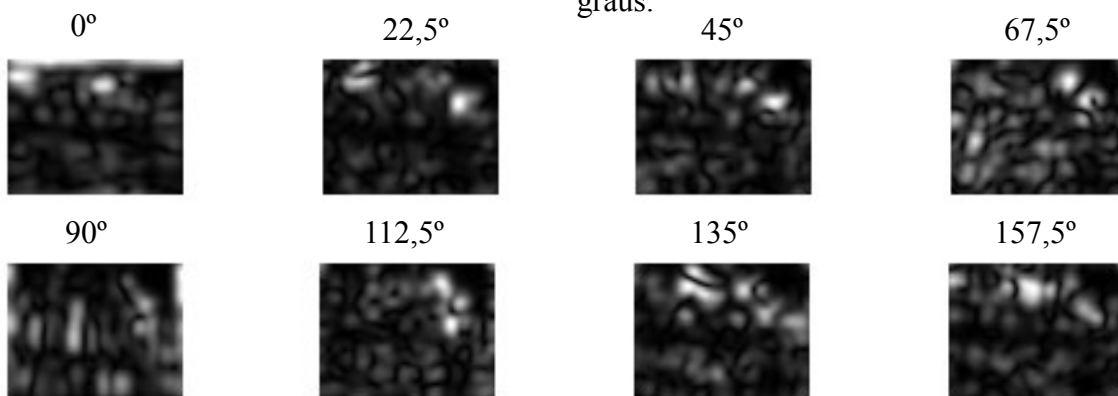
Na segunda etapa, a extração de características é obtida pela convolução entre a imagem em níveis de cinza e o banco de filtros originado pela representação de wavelet Gabor. Nessa representação, esse banco possui cinco escalas diferentes, oito orientações e conforme a Equação (2.3) a coordenada do pixel (x, y) é considerada $(39, 39)$ (HAGHIGHAT; ZONOUZ; ABDEL-MOTTALEB, 2013). A Figura 4.3 ilustra o módulo do resultado dessa convolução. Essa figura mostra as 40 sub-imagens obtidas utilizando essas escalas e essas orientações.

A primeira e segunda linha da Figura 4.3 ilustra 8 sub-imagens obtidas utilizando a frequência de 0,25 e oito orientações (0° ; $22,5^\circ$; 45° ; $67,5^\circ$; 90° ; $112,5^\circ$; 135° e $157,5^\circ$). Cada uma das outras linhas dessa figura é obtida usando a frequência encontrada na Equação (2.4) e as mesmas orientações utilizadas nessa primeira linha.

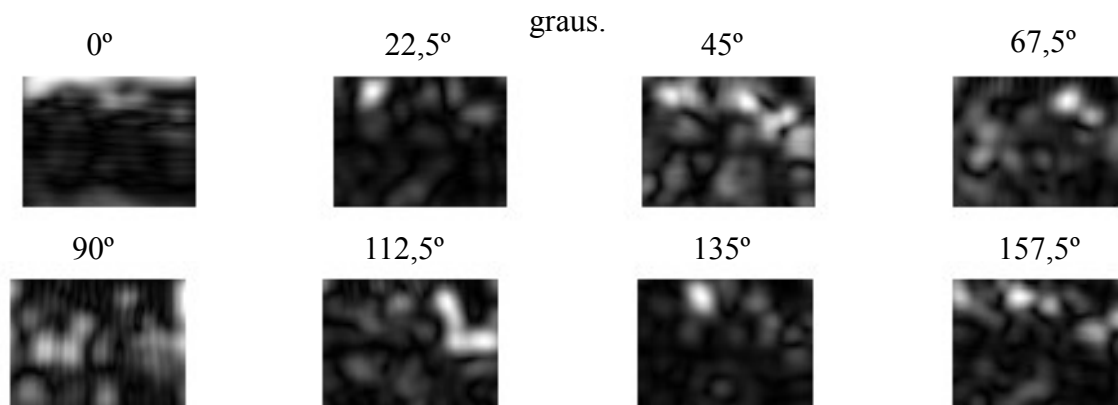
Figura 4.2 - Módulo do resultado obtido da convolução entre a imagem em níveis de cinza e a representação de wavelet Gabor.



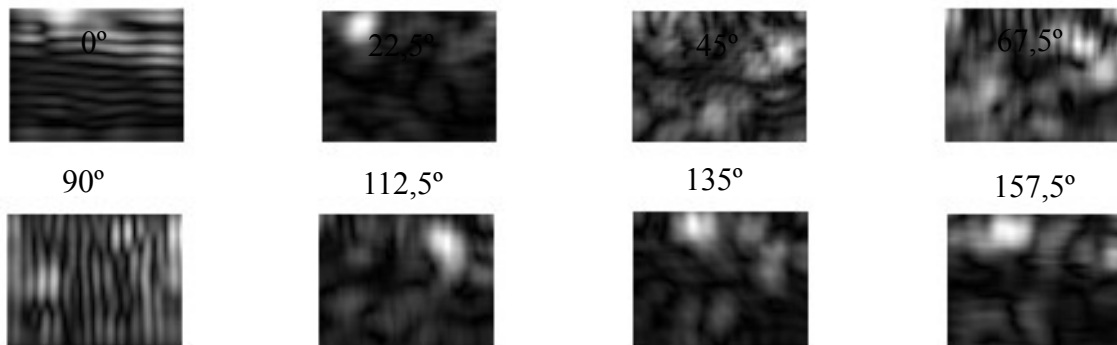
(a) Extração das características com frequência igual a 0,25 e oito orientações em graus.



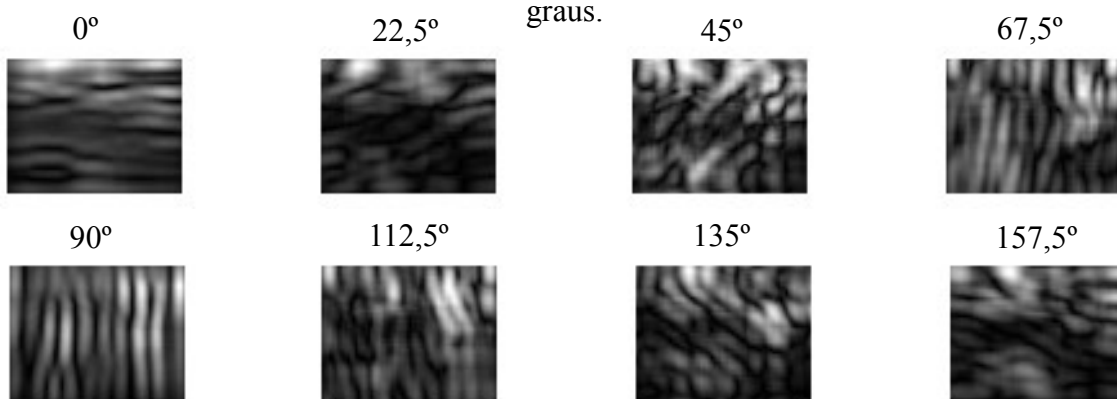
(b) Extração das características com frequência igual a 0,1768 e oito orientações em



(c) Extração das características com frequência igual a 0,1250 e oito orientações em graus.



(d) Extração das características com frequência igual a 0,0884 e oito orientações em graus.



(e) Extração das características com frequência igual a 0,0625 e oito orientações em graus.

As linhas e as colunas das imagens da Figura 4.3 são sub-amostradas por um fator de redução igual a 4. O vetor de características é obtido rearranjando o resultado dessa sub-amostragem em um vetor coluna. Finalmente, para ser utilizado na rede MLP, o vetor de características é normalizado com média zero e variância igual a um. Porém, para uma convergência mais rápida da rede, a dimensionalidade do vetor normalizado é reduzida.

A dimensão do vetor de características é obtida pela Equação (2.7). Como as imagens utilizadas nesse trabalho são 160 x 120 pixels e na representação de

wavelet Gabor foram usadas 5 escalas de frequência e 8 orientações, a dimensão deste vetor é igual a 48.000 amostras.

4.4 - Redução da dimensionalidade

Com a finalidade de reduzir a dimensionalidade do vetor de características aplicou-se a transformada discreta wavelet nesse vetor até o nível 3. O resultado dessa operação é o novo vetor de características obtido pelo coeficiente de aproximação 3 da TW (cA_3). O tamanho deste novo vetor depende do tipo e da ordem da wavelet utilizada. Por exemplo, se é usada a transformada wavelet de Daubechies de primeira ordem, o tamanho do vetor de características é igual a 6000 amostras. É esse vetor que é utilizado para o treinamento e os testes da rede MLP.

4.5 - Treinamento e teste da rede MLP

Para realizar o reconhecimento dos edifícios foi utilizada uma rede neural artificial MLP com uma única camada escondida. Nessa rede, o número de neurônios da camada de entrada é igual ao tamanho do vetor de características. Por meio da validação cruzada, foi determinado o número de 200 neurônios na camada escondida. Entretanto, para a camada de saída, optou-se pela adoção de vetores-alvo do tipo bipolares e ortogonais (VBO)s. Esses vetores são obtidos por meio de um algoritmo e sua dimensão é sempre igual a uma potência de 2. Como estamos utilizando 21 classes, precisa-se de VOB's de tamanho 32. As Equações (4.1) e (4.2) mostram as definições dos vetores binários e bipolares, respectivamente.

$$\vec{V}_{ij} = \begin{cases} 1 & \text{se } i = j \\ 0 & \text{se } i \neq j \end{cases} \quad (4.1)$$

$$\vec{V}_{ij} = \begin{cases} 1 & \text{se } i = j \\ -1 & \text{se } i \neq j \end{cases} \quad (4.2)$$

Manzan (2012) mostrou que o uso de vetores bipolares ortogonais (VBO)s aumenta a distância euclidiana entre os pontos de saída da rede. Esse aumento beneficia o mapeamento da rede MLP, resultando em taxas de acerto maiores e menor tempo de treinamento.

Em Fausset (1994) foi proposto o algoritmo para gerar os VBOs. Esse algoritmo utiliza um vetor de sementes que gera uma sequência de vetores até a determinação final dos VBOs. O número de componentes de cada VBO depende do número de vetores desejados e dos componentes escolhidos para o vetor semente. A Equação (4.3) calcula o número de componentes de cada vetor.

$$n = 2^k m \quad (4.3)$$

onde:

2^k – número de vetores ortogonais entre si para $k > 0$; e

m – número de componentes em um vetor de sementes.

Os passos deste algoritmo são:

Passo 1 – Conforme a aplicação, os valores de m e k devem ser determinados pelo usuário. O valor de m inicia igual a 1 e a medida que ele aumenta, a quantidade de VBO's cresce.

Passo 2 – O vetor de sementes é obtido pela Equação (4.4). É importante ressaltar que esse vetor pode ter um único componente.

$$V^0_m = \overbrace{(1, 1, \dots, 1)}^m{}^T \quad (4.4)$$

Passo 3 – O número de componentes do VBO é obtido pela Equação (4.3).

Passo 4 – A partir do vetor de sementes obtido no passo 3 são construídos os vetores $V^1_{2m} = fcc(V^0_m, V^0_m)$ e $V^2_{2m} = fcc(V^0_m, -V^0_m)$.

onde:

$fcc(V_1, V_2)$ – função de concatenação dos vetores V_1 e V_2 .

É importante observar que o vetor de sementes é utilizado duas vezes. Na primeira é realizada a concatenação de V_1 com V_2 . Na segunda vez, o sinal de V_2 é alterado.

Passo 5 – A partir dos vetores V_1 e V_2 obtidos no passo 4 com a utilização da função de concatenação são determinados quatro novos vetores. Estes vetores são:

$$V_{4m}^1 = fcc(V_{2m}^1, V_{2m}^1), \quad V_{4m}^2 = fcc(V_{2m}^1, -V_{2m}^1), \quad V_{4m}^3 = fcc(V_{2m}^2, V_{2m}^2) \quad \text{e}$$

$$V_{4m}^4 = fcc(V_{2m}^2, -V_{2m}^2).$$

Os vetores determinados no passo 4 são utilizados duas vezes na função de concatenação. Na primeira o sinal do segundo argumento é mantido e na segunda vez ele é alterado.

Passo 6 – As concatenações são realizadas até obter 2^k vetores ortogonais com n componentes $(V_n^1, \dots, V_n^{2^k})$.

No treinamento e nos testes com a MLP foram utilizadas as 80 primeiras imagens das 21 categorias do banco de imagens de edifícios, totalizando 1680 amostras. Deste total, 1344 (64 imagens de cada categoria) foram utilizadas para o treinamento da rede e 336 (16 imagens de cada categoria) para teste. É importante ressaltar que o conjunto de amostras utilizado no treinamento é diferente do conjunto de amostras usado nos testes.

O critério de parada do treinamento da MLP foi o alcance do erro quadrático médio tolerado. Após realizar vários testes, foram consideradas as taxas de erro tolerável e aprendizagem iguais a 0,001 e 0,01, respectivamente. Para acelerar a convergência da rede foi utilizado no treinamento o termo momentum e a taxa de aprendizagem adaptativa.

4.6 - Considerações finais deste capítulo

Esse capítulo apresentou o método proposto nessa tese.

O próximo capítulo apresenta os resultados obtidos nos testes realizados.

E finalmente, são realizadas as conclusões destes resultados.

CAPÍTULO 5

5. RESULTADOS OBTIDOS

5.1 - Introdução

Esse capítulo apresenta os resultados obtidos utilizando o sistema de reconhecimento de edifícios desenvolvido nessa tese. Finalmente, são realizadas as conclusões destes resultados.

Os testes foram realizados para verificar o comportamento da rede neural de múltiplas camadas (MLP) com a variação dos tipos de transformadas wavelets “mãe” utilizadas na redução da dimensionalidade e na variação do número de orientações do filtro de Gabor.

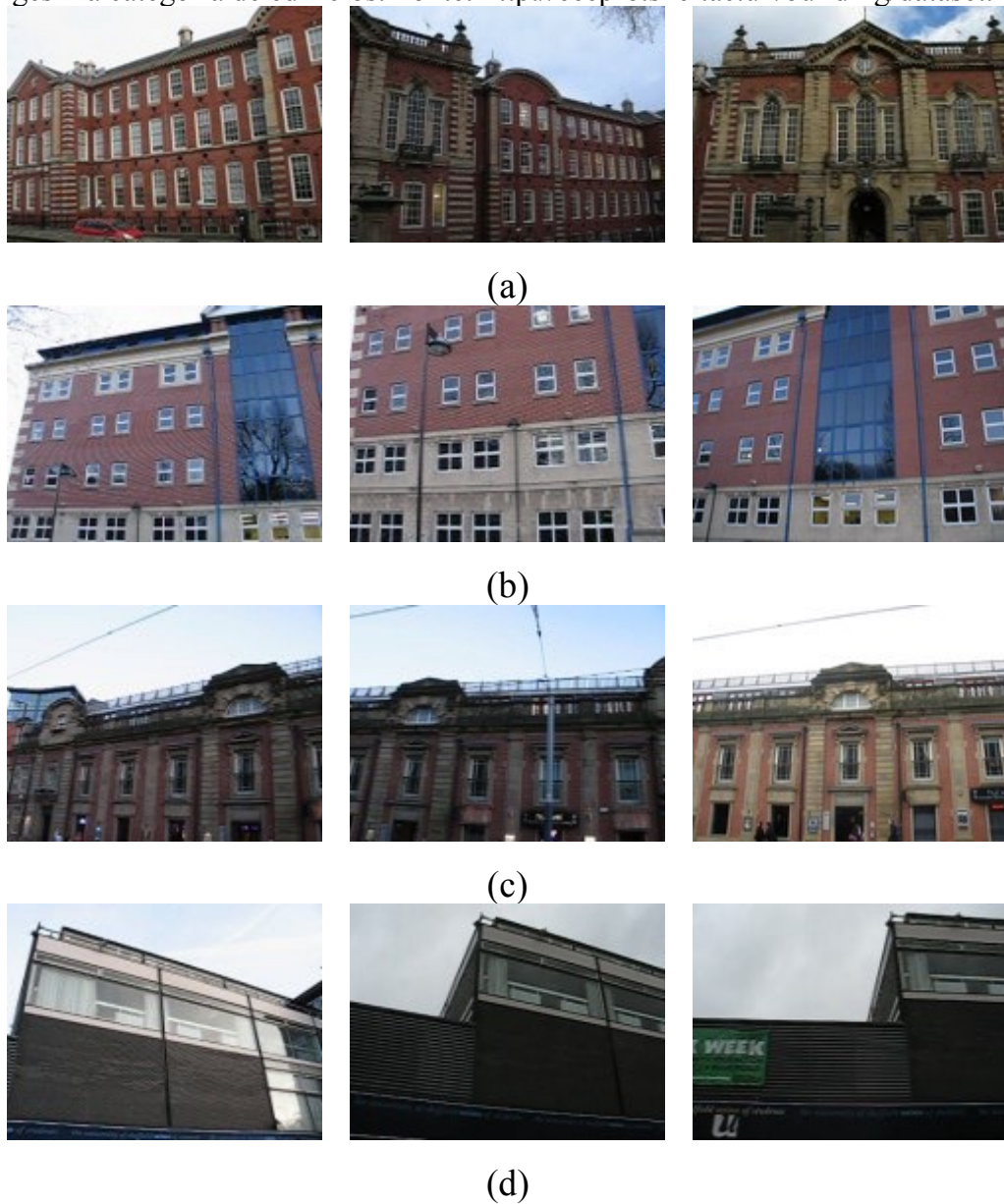
5.2 - Banco de dados

Para avaliar o desempenho do método proposto, os testes foram realizados utilizando um banco de dados de imagem da Universidade de Sheffield (SBID) (<http://eeepro.shef.ac.uk/building/dataset.rar>). Nesse banco existem 40 categorias de edifícios, totalizando 3.192 imagens no formato JPEG com resolução de 160 x 120 pixels. Todas as imagens deste banco foram tomadas em torno do centro da cidade da Universidade de Sheffield na Inglaterra, usando uma câmera digital Cannon G90, incluindo fotos e clipes de vídeo. Essas imagens foram capturadas sob diferentes variações da imagem em relação a

rotação, dimensionamento, diferentes condições de iluminação, oclusões e vibração (LI e ALLINSON, 2013).

A Figura 5.1 ilustra algumas categorias de imagens de edifícios utilizadas nesse trabalho.

Figura 5.1 - (a) imagens da primeira categoria, (b) imagens da terceira categoria, (c) imagens da sexta categoria, (d) imagens da décima sexta categoria, (e) imagens da vigésima categoria de edifícios. Fonte: <http://eeepro.shef.ac.uk/building/dataset.rar>.





(e)

Na Figura 5.1 na linha (a), é possível verificar que um ser humano não seria capaz de identificar que essas imagens mostradas se referem ao mesmo edifício.

5.3 - Resultados obtidos

A Tabela 5.1 mostra os recursos computacionais utilizados nesse trabalho.

Tabela 5.1- Recursos computacionais usados nesse trabalho.

Processador	IntelCorei5
Memória RAM	4G
Disco rígido	500G
Ferramenta de programação	MatlabR2013b

Com a finalidade de reduzir o tempo de processamento dos testes realizados foram usadas das 40 categorias do banco de dados apenas 21. Para esses testes, utilizou-se apenas as primeiras 80 imagens das 21 categorias. Dessas imagens, 64 (80%) foram usadas para o treinamento da rede MLP e 16 (20%) para teste. A configuração dessa rede foi gradiente descendente com o termo momentum igual a 0,8 e a taxa de aprendizagem adaptativa, 200 e 32

neurônios nas camadas intermediária e de saída, respectivamente e o erro desejado de 0,001.

Para verificar o comportamento da MLP em relação a transformada wavelet, foram realizadas três simulações com as TW de Daubechies, Coiflet e Symlet, usando o coeficiente de aproximação de nível 3 (cA3). Nessas simulações, a extração das características foram feitas utilizando o filtro de Gabor com 8 orientações, 5 escalas de frequências e um fator de redução de linhas e colunas igual a 4.

O primeiro teste foi realizado variando a ordem da transformada wavelet de Daubechies de 1 a 10. A Tabela 5.2 mostra a taxa de acerto da MLP nesse teste.

Tabela 5.2 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Daubechies usando os coeficientes de aproximação de nível 3.

TW de Daubechies	Taxa de reconhecimento %	Nº de Épocas	Nº de Características
db1	94	220	6000
db2	94	232	6002
db3	93	157	6004
db4	92	227	6006
db5	95	227	6007
db6	93	157	6009
db7	94	156	6011
db8	93	157	6013
db9	93	233	6014
db10	93	229	6016

Pode-se observar na Tabela 5.2, que a maior taxa de acerto foi de 95% para Daubechies de ordem 5 e para as outras ordens essas taxas variaram de 92 a 94%.

O segundo teste foi realizado variando a ordem da transformada wavelet Coiflet de 1 a 5. A Tabela 5.3 ilustra a taxa de acerto da MLP nesse teste.

Tabela 5.3 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Coiflet usando os coeficientes de aproximação de nível 3.

TW de Coiflet	Taxa de reconhecimento %	Nº de Épocas	Nº de características
coif1	93	157	6004
coif2	94	157	6009
coif3	93	160	6014
coif4	93	157	6020
coif5	94	229	6025

Analisando a Tabela 5.3, verifica-se que tanto a coiflet de ordem 2 quanto a de ordem 5, apresentaram o mesmo resultado, ou seja, a taxa de acerto foi de 94%. Porém em relação ao número de épocas e uma quantidade menor de características, o melhor resultado foi a coiflet de ordem 2.

De forma equivalente, o terceiro teste considerou os mesmos parâmetros usados no primeiro e segundo testes. A diferença foi utilizar a transformada wavelet symlet. A Tabela 5.4 mostra os resultados obtidos usando essa transformada para as ordens de 1 a 10.

Tabela 5.4 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Symlet usando os coeficientes de aproximação de nível 3.

TW de Symlet	Taxa de reconhecimento %	Nº de Épocas	Nº de características
sym1	94	157	6000
sym2	93	157	6002
sym3	93	227	6004
sym4	93	157	6006
sym5	93	223	6007
sym6	93	227	6009
sym7	94	156	6011
sym8	91	157	6013
sym9	92	160	6014
sym10	93	157	6016

Verificando os resultados mostrados na Tabela 5.4, nota-se que o melhor resultado em relação a taxa de acerto foi obtido ao utilizar a Symlet de ordem 7. Esse resultado foi igual a 94% em 156 épocas.

Comparando as taxas de reconhecimento das Tabelas 5.2 a 5.4, verifica-se que a redução de dimensionalidade usando a transformada wavelet de Daubechies de ordem 5 apresentou a melhor taxa de reconhecimento da rede neural.

No quarto teste foi usado os mesmos parâmetros adotados nos outros testes, a única diferença, é a utilização dos coeficientes de detalhe (cD) de nível 3 (cD3). Nesse teste, o comportamento da MLP foi verificado, variando a ordem da transformada wavelet de Daubechies de 1 a 10, mantendo as 5 escalas de frequências, 8 orientações e um fator de redução de linhas e colunas igual a 4. A Tabela 5.5 mostra a taxa de acerto da MLP nesse teste.

Tabela 5.5 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Daubechies usando os coeficientes de detalhe de nível 3.

TW de Daubechies	Taxa de reconhecimento %	Nº de Épocas	Nº de Características
db1	92	142	6000
db2	89	138	6002
db3	90	136	6004
db4	89	136	6006
db5	92	140	6007
db6	87	137	6009
db7	87	139	6011
db8	87	138	6013
db9	87	137	6014
db10	87	138	6016

Pode-se observar na Tabela 5.5, que a maior taxa de acerto foi de 92% para Daubechies de ordem 1 e ordem 5 e para as outras ordens essas taxas variaram de 87 a 90%. Nesse teste, considera-se a Daubechies de ordem 5 como a melhor taxa de acertos devido ao número de épocas melhor.

Dando sequência nos testes, na quinta simulação foi usada a família coiflet variando a ordem de 1 a 5 com os mesmos parâmetros adotados na quarta simulação. Os resultados obtidos nesse experimento é mostrado na Tabela 5.6.

Tabela 5.6 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Coiflets usando os coeficientes de detalhe de nível 3.

TW de Coiflet	Taxa de reconhecimento %	Nº de Épocas	Nº de características
coif1	89	139	6004
coif2	88	136	6009
coif3	88	142	6014
coif4	88	138	6020
coif5	88	137	6025

Analisando a Tabela 5.6, verifica-se que o melhor resultado foi de 89% utilizando a coiflet de ordem 1.

De modo análogo, o sexto teste usou os mesmos parâmetros do quarto e quinto testes. Nessa simulação utilizou a symlet variando a ordem de 1 a 10. A taxa de acertos da MLP para esse teste é ilustrada na Tabela 5.7.

Tabela 5.7 - Taxa de reconhecimento da MLP para diferentes ordens da TW de Symlets usando os coeficientes de detalhe de nível 3.

TW de Symlet	Taxa de reconhecimento %	Nº de Épocas	Nº de características
sym1	91	136	6000
sym2	89	136	6002
sym3	89	138	6004
sym4	89	137	6006
sym5	88	139	6007
sym6	88	140	6009
sym7	88	139	6011

sym8	88	142	6013
sym9	87	140	6014
sym10	86	140	6016

No sexto teste o melhor resultado foi de 91% utilizando a symlet de ordem 1 e para as outras ordens essas taxas variaram de 86 a 89% como ilustrou a Tabela 5.7.

Observando os resultados obtidos nas Tabelas 5.5 a 5.7, verifica-se que o melhor resultado usando o coeficiente de detalhe 3 foi com a utilização da transformada wavelet de Daubechies de ordem 5. A taxa de acertos foi de 92%.

Com a finalidade de comparar os coeficientes de aproximação e detalhe de nível 3 em relação as taxas de acertos, as Figuras 5.2, 5.3 e 5.4 mostram respectivamente, os gráficos comparativos das famílias Daubechies, Coiflet e Symlet variando as suas ordens em relação aos coeficientes de aproximação e detalhe de nível 3 (cA3 e cD3).

Figura 5.2 – Gráfico comparativo entre cA3 e cD3 para Daubechies com a ordem variando de 1 a 10.

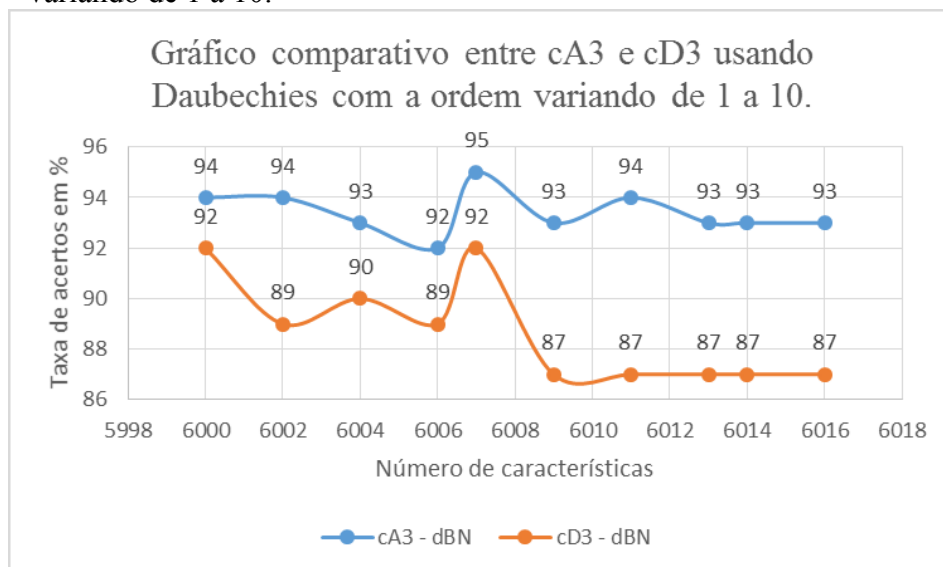


Figura 5.3 – Gráfico comparativo entre cA3 e cD3 para Coiflet com a ordem variando de 1 a 5.

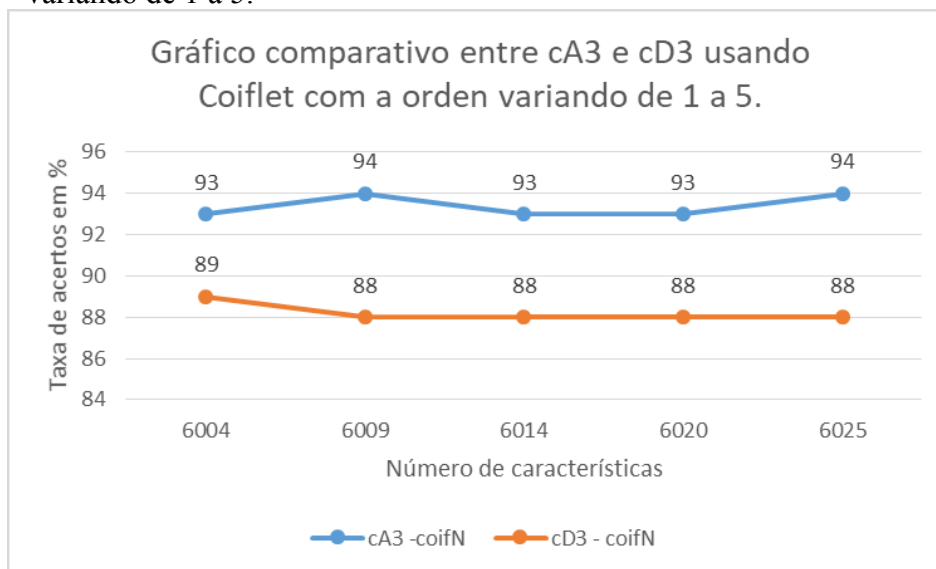
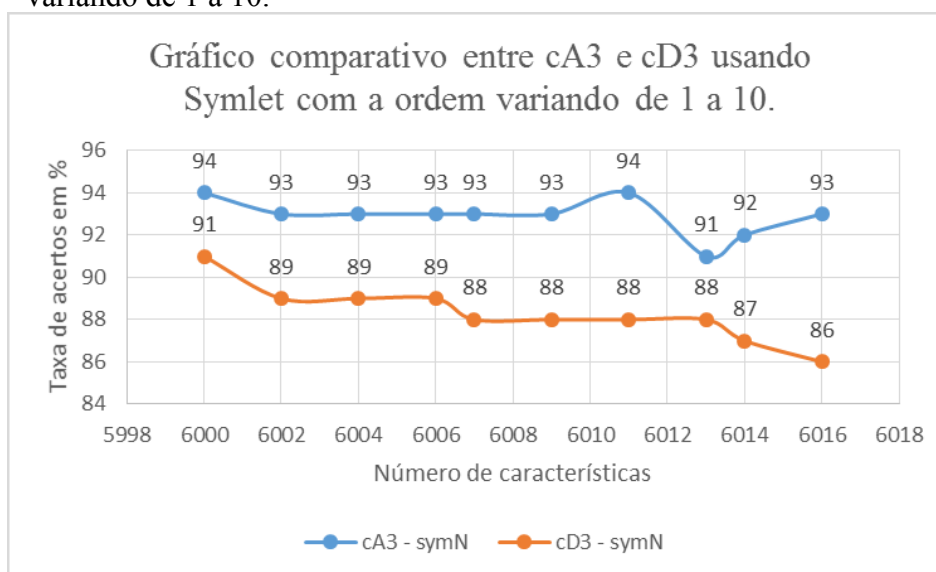


Figura 5.4 – Gráfico comparativo entre cA3 e cD3 para Symlet com a ordem variando de 1 a 10.



Analisando os resultados das Figuras 5.2 a 5.4, verifica-se que o coeficiente de aproximação nível 3 obteve o melhor resultado para todos os testes realizados em relação ao coeficiente de detalhe.

No sétimo teste, o comportamento da MLP foi analisado variando o número de orientações do filtro de Gabor, mantendo as 5 escalas de frequência,

um fator de redução de linhas e colunas igual a 4 e utilizando o coeficiente de aproximação 3. Após os testes realizados na primeira, segunda e terceira simulação, adotou-se neste experimento a transformada wavelet de Daubechies de ordem 5 como a técnica para reduzir a dimensionalidade. A Tabela 5.8 mostra a taxa de reconhecimento da MLP nessa simulação.

Tabela 5.8 - Taxa de reconhecimento da MLP variando o parâmetro de orientação do filtro de Gabor para TW de Daubechies de ordem 5 usando o coeficiente de aproximação.

Nº de orientações θ	Taxa de reconhecimento %	Nº de Épocas	Nº de características
1	94	715	757
2	93	933	1507
3	91	207	2257
4	96	379	3007
5	93	160	3757
6	94	230	4507
7	94	161	5257
8	95	227	6007

Pode-se observar na Tabela 5.5, que a maior taxa de reconhecimento foi de 96% usando 4 orientações na extração de características. Para as outras orientações essas taxas variaram de 91 a 95%.

No oitavo teste foi utilizada a mesma metodologia em relação ao sétimo. Nesse teste foi mantido o fator de redução de linhas e colunas igual a 4, as 5 escalas de frequência, variando o número de orientações do filtro de Gabor (1 a 8) e utilizando o coeficiente de detalhe 3. Após os testes realizados na quarta, quinta e sexta simulação, adotou-se neste experimento a transformada wavelet

de Daubechies de ordem 5 como a técnica para reduzir a dimensionalidade por apresentar o melhor resultado em relação a taxa de acertos. A Tabela 5.9 mostra a taxa de reconhecimento da MLP nessa simulação.

Tabela 5.9 - Taxa de reconhecimento da MLP variando o parâmetro de orientação do filtro de Gabor para TW de Daubechies de ordem 5 usando o coeficiente de detalhe.

Nº de orientações θ	Taxa de reconhecimento %	Nº de Épocas	Nº de características
1	85	1326	757
2	89	699	1507
3	89	681	2257
4	89	147	3007
5	86	143	3757
6	89	142	4507
7	90	139	5257
8	88	137	6007

Analisando os resultados da Tabela 5.9, verifica-se que a maior taxa de acertos foi de 90% usando 7 orientações, 5 escalas de frequência e fator de redução de linhas e colunas igual a 4. As demais orientações variaram de 85 a 89%.

Portanto, é possível verificar que o coeficiente de aproximação obteve melhores resultados em relação ao coeficiente de detalhe. Isso pode ser verificado ao comparar as Tabelas 5.8 e 5.9, respectivamente.

Com a finalidade de reduzir o tamanho do vetor de características, no nono teste, utilizou a transformada wavelet de Daubechies de ordem 5 como técnica para redução de dimensionalidade, um fator de redução de linhas e

colunas igual a 10, variando o número de orientações do filtro de Gabor (1 a 8) e mantendo as 5 escalas de frequência. Nesse teste foi usado os coeficientes de aproximação (cA) de nível 3. A Tabela 5.10 ilustra esses resultados.

Tabela 5.10 - Taxa de reconhecimento da MLP variando o parâmetro de orientação do filtro de Gabor para TW de Daubechies de ordem 5 usando o coeficiente de aproximação com fator de redução de linha e coluna igual a 10.

Nº de orientações θ	Taxa de reconhecimento %	Nº de Épocas	Nº de características
1	82	1184	127
2	78	2915	247
3	81	281	367
4	79	739	487
5	81	329	607
6	82	378	727
7	84	309	847
8	81	260	967

Nesse teste de acordo com a Tabela 5.10 é possível verificar que com a redução das características, o coeficiente de aproximação de nível 3 (cA3), tem a sua taxa de acertos menor, quando comparado com os resultados da Tabela 5.8. Assim, a maior taxa de acertos foi de 84% com 7 orientações. Os demais resultados variaram de 79 a 82%.

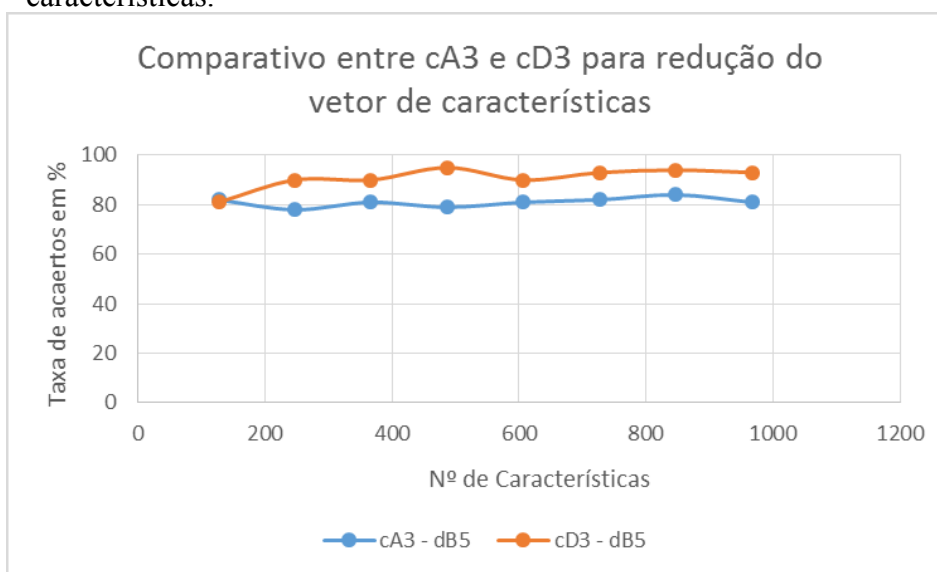
Seguindo a mesma metodologia do nono teste, o décimo teste usou os coeficientes de detalhe de nível 3 (cD3) ao invés dos coeficientes de aproximação de nível 3 (cA3). Os resultados são mostrados na Tabela 5.11.

Tabela 5.11 - Taxa de reconhecimento da MLP variando o parâmetro de orientação do filtro de Gabor para TW de Daubechies de ordem 5 usando o coeficiente de detalhe com fator de redução de linha e coluna igual a 10.

Nº de orientações θ	Taxa de reconhecimento %	Nº de Épocas	Nº de características
1	81	1733	127
2	90	764	247
3	90	192	367
4	95	210	487
5	90	167	607
6	93	169	727
7	94	160	847
8	93	165	967

Nesse teste a maior taxa de acertos foi de 95% com 4 orientações, 5 escalas de frequências, redução de linhas e colunas igual a 10 e utilização da transformada wavelet de Daubechies de ordem 5 para redução do vetor de características. Percebe-se ainda por este experimento, que se o número de características diminui, o coeficiente de detalhe é melhor em relação a taxa de acertos quando comparado com o coeficiente de aproximação. Assim, a Figura 5.5 ilustra esse comportamento ao comparar os resultados obtidos nas Tabelas 5.10 e 5.11, respectivamente.

Figura 5.5 – Gráfico comparativo entre cA3 e cD3 com redução do número de características.



Analizando a Figura 5.5 é visível que o coeficiente de detalhe apresenta melhores resultados quando comparados com o coeficiente de aproximação. Diante disso, entende-se que quanto menor o número de características usados no reconhecimento, a imagem vai perdendo sua qualidade ou a sua aproximação, assim os coeficientes de detalhe prevalece para distinguir um edifício do outro.

A Tabela 5.12 mostra os tempos de processamento gasto no treinamento e no teste da rede neural MLP que apresentou a maior taxa de reconhecimento, ou seja, 96% com fator de redução de linhas e coluna igual a 4, 5 escalas de frequência, 4 orientações e o uso da transformada wavelet de Daubechies de ordem 5 na redução de dimensionalidade, sendo o vetor de características dado pelo coeficiente de aproximação de nível 3. Os testes para verificar esses tempos foram feitos utilizando 100, 200 e 400 neurônios na camada escondida.

Tabela 5.12 - Tempos de processamentos gastos no treinamento e no teste da rede neural MLP para o melhor resultado usando o coeficiente de aproximação de nível 3.

Nº de neurônios na camada escondida	Taxa de reconhecimento %	Tempo de treinamento	Tempo de teste
100	93%	4min29s	16s
200	96%	2min22s	21s
400	96%	2min13s	34s

Pode-se verificar na Tabela 5.12 que os tempos de processamentos gastos no treinamento e no teste das MLP's com 100, 200 e 400 neurônios na camada escondida apresentaram valores muito próximos. Porém, as MLP's que apresentaram melhores taxas de reconhecimento foram as que tinham 200 e 400 neurônios na camada escondida. Assim, em todos os testes utilizou-se nas MLP's o menor número de neurônios nessa camada, ou seja, 200 neurônios na camada escondida.

Utilizando a mesma analogia, também foi realizado o teste para verificar os tempos de processamento para o melhor resultado quando o número de características foi reduzido por um fator de subamostragem igual a 10 e utilização do coeficiente de detalhe de nível 3 como vetor de características. Esse resultado foi de 95% conforme foi mostrado na Tabela 5.11. Os experimentos também foram conduzidos com 100, 200 e 400 neurônios na camada escondida. Os resultados são mostrados na Tabela 5.13.

Tabela 5.13 - Tempos de processamentos gastos no treinamento e no teste da rede neural MLP para o melhor resultado usando o coeficiente de detalhe de nível 3.

Nº de neurônios na camada escondida	Taxa de reconhecimento %	Tempo de treinamento	Tempo de teste
100	95	6s	4s

200	95	7s	4s
400	95	13s	6,9s

Comparando as Tabelas 5.13 em relação a 5.12 é notável que a redução de característica reduz consideravelmente os tempos de processamentos gastos no treinamento e teste.

A Tabela 5.14 mostra o melhor resultado obtido utilizando o método proposto nesse trabalho e os resultados encontrados na literatura.

Tabela 5.14 - Resultado obtido utilizando o método proposto nesse trabalho e os resultados encontrados na literatura.

TRABALHOS CORRELATOS	BASE DE DADOS	TAXA DE RECONHECIMENTO %
Método proposto	SBID	96
BPBR (LI e ALLINSON, 2009)	SBID	85,3
RFBR (LI e ALLINSON, 2010)	SBID	93
Groeneweg et al. (2006)	ZuBud	91
<i>Perceptual grouping</i> (IQBAL, e AGGARWAL, 1999)	150 building images	83,7
<i>Consistent line clusters</i> (LI e SHAPIRO, 2002)	977 Color images	94,2
<i>Hierarchical building recognition</i> (ZHANG, et al. 2007)	ZuBud	95
<i>Rancking scheme</i> (PHILBIN, et al. 2007)	OBD (5 M)	95,3
Zheng et al. (2009)	5312 Landmarks from 1259 cities	80,8
<i>Sketch-based representations</i> (CHUNG; HAN; HE, 2009)	ZuBud	81

Fonte: Li, et al. (2014)

Pode-se verificar na Tabela 5.7 que o método proposto deste trabalho apresentou a melhor taxa de reconhecimento.

5.4 - Conclusões

Esse capítulo apresentou os resultados obtidos utilizando o sistema de reconhecimento de edifícios desenvolvido nessa tese.

Nos seis primeiros testes realizados, as transformadas wavelet utilizadas foram Daubechies, Coiflet e Symlet. Sendo que nos três primeiros utilizou-se os coeficientes de aproximação na redução da dimensionalidade de nível 3 e os outros três, os coeficientes de detalhe também de nível 3. Destes testes, verificou-se que a aplicação da transformada wavelet de Daubechies de ordem 5 foi a que apresentou a melhor taxa de reconhecimento, sendo então esse tipo de wavelet usado nas demais simulações.

No sétimo teste, foi utilizado como técnica de redução de dimensionalidade a transformada wavelet de Daubechies ordem 5 e o número de orientações do filtro de Gabor variando de 1 a 8. A melhor taxa de reconhecimento nesse teste foi 96% usando 4 orientações do filtro de Gabor.

Com a finalidade de reduzir o tamanho do vetor de características, no nono teste, utilizou a transformada wavelet de Daubechies de ordem 5, um fator de redução de linhas e colunas igual a 10, variando o número de orientações do filtro de Gabor (1 a 8) e mantendo as 5 escalas de frequência. Na comparação destes testes em relação ao coeficiente de aproximação e detalhe, percebe-se que quanto menor o número de características usados no reconhecimento, a imagem vai perdendo sua qualidade ou a sua aproximação, assim os coeficientes de

detalhe prevalecem para distinguir um edifício do outro. Neste teste o melhor resultado foi a taxa de reconhecimento de 95% com 487 características.

O próximo capítulo apresenta as conclusões e as contribuições deste trabalho.

CAPÍTULO 6

6. CONCLUSÕES GERAIS, CONTRIBUIÇÕES E TRABALHOS FUTUROS

6.1 - Introdução

Esse capítulo apresenta as conclusões, as contribuições deste trabalho e os trabalhos futuros. Finalmente, são realizadas as considerações finais deste capítulo.

6.2 - Conclusões

Esse trabalho apresentou o estudo de técnicas de extração de características, redução de dimensionalidade e a rede MLP utilizadas para o reconhecimento de edifícios. O banco de imagens usado nos testes possui características diferentes, como por exemplo: dimensionamento, iluminação, ângulos diferentes, obstruções, dentre outras. Os testes foram realizados com a finalidade de verificar como a redução de dimensionalidade influencia no comportamento da rede MLP.

Os testes foram realizados variando as ordens das transformadas wavelet de Daubechies, Symlet e coiflet. Os resultados obtidos nesses testes mostraram que as taxas de reconhecimento apresentaram praticamente os mesmos valores. Porém, a transformada wavelet de Daubechies de ordem 5 foi a que apresentou a maior taxa de reconhecimento. Assim, passou-se a utilizá-la nos demais testes.

Outra forma utilizada para verificar o comportamento da MLP foi variar as orientações do filtro de Gabor. Nos testes realizados com essa variação a maior taxa de reconhecimento obtida foi de 96% usando a transformada wavelet de Daubechies de ordem 5 e quatro orientações do filtro de Gabor. Essa taxa foi o melhor resultado encontrado nesse trabalho.

Com o objetivo de reduzir a dimensionalidade do vetor de características para verificar o comportamento da MLP, usou-se uma subamostragem com fator de 10 na filtragem de Gabor e a wavelet de Daubechies de ordem 5 para reduzir ainda mais o vetor de características. Assim foi possível verificar que o uso dos coeficientes de detalhe apresentam resultados melhores em relação aos coeficientes de aproximação, sendo o melhor resultado de 95% com 418 características.

6.3 - Contribuições

As contribuições deste trabalho são:

- Utilização apenas da característica de textura com a utilização do filtro de Gabor, visto que outros trabalhos trabalham com muitas características extraídas de técnicas diferentes;
- Aplicação da transformada wavelet na redução de dimensionalidade; e
- Utilização de rede neural artificial do tipo MLP no reconhecimento de edifícios e parametrização dessa rede nessa aplicação.

6.4 - Trabalhos Futuros

A área de reconhecimento de edifícios se mostrou bastante desafiadora. Assim como trabalhos futuros propõe-se:

- Usar as características de cores dos edifícios para o reconhecimento;
- Utilização de redes de aprendizado profundo (“*Deep Learning*”);
- Testar outros classificadores neste mesmo trabalho, como por exemplo, o Softmax; e
- Testar outros tipos de wavelets na redução da dimensionalidade do vetor de características.

6.5 - Considerações finais deste capítulo

Esse capítulo apresentou as conclusões, as contribuições deste trabalho e os trabalhos futuros.

REFERÊNCIAS BIBLIOGRÁFICAS

- ALVES, D. M. **Quantificação da Glicemia através de análises de imagens da íris humana utilizando redes neurais**. Dissertação (Mestrado) – Universidade de São Paulo, Escola de Engenharia de São Carlos. São Carlos – SP: 2007.
- ANGELO, N. P. **Aplicação de Filtros de Gabor no Processo de Classificação de Imagens Digitais com Base em Atributos de Textura**. Dissertação (Mestrado) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2000.
- BRAGA, A. P.; CARVALHO, A. P. L. F.; LUDEMIR, T. B. **Redes Neurais Artificiais** - Teoria e Aplicações. 2. ed. Rio de Janeiro: LTC, 2007.
- CASSIMIRO, I. J. G. **Identificação de Falhas em Máquinas Elétricas Rotativas Usando Sistemas Inteligentes**. Monografia (Graduação em Engenharia com ênfase em Sistemas de Energia e Automação) – Universidade de São Paulo, Escola de Engenharia de São Carlos. São Carlos – SP: 2013.
- CHUNG, Y, C., HAN, T. X., HE, Z. Building recognition using sketch-based representations and spectral graph matching, **in Proc. IEEE Int. Conf. Comput. Vis.** p. 2014-2020, 2009.
- DUAN, W., ALLINSON, N. M. Vanishing Points Detection and Line Grouping for Complex Building Façade Identification, **WSCG**, p. 213-220, 2010.

- FAUSETT, L. *Fundamentals of neural networks: Architecture, algorithms, and applications*. Prentice-Hall, 1994.
- FERREIRA, J. C. **Utilização da transformada Wavelet para detectar variações anormais de frequência em sistemas de geração distribuída**. Dissertação (Mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica. Uberlândia – MG: 2009.
- GROENEWEG, N. et al. A fast offline building recognition application on a mobile telephone, **Advanced Concepts for Intelligent Vision Systems**, p. 1122-1132, 2006.
- HAGHIGHAT, M., ZONOUZ, S., ABDEL-MOTTALEB, M. Identification Using Encrypted Biometrics, **Computer Analysis of Images and Patterns, Springer Berlin Heidelberg**, v. 8048, p. 440-448, 2013.
- HAYKIN, S. **Redes Neurais: Princípios e Prática**. 8. ed. Porto Alegre: Bookman, 2007.
- HEBB, D. *The organization behavior: A Neuropsychological Theory*. New York, NY, USA: John Wiley & Sons, 1949.
- HUTCHINGS, R. J., MAYOL, W, W. Building Recognition for Mobile Devices: Incorporating Positional Information with Visual Features, **CSTR-06-017**, Computer Science, University of Bristol, 2005.
- KOVÁCS, Z. L. **Redes Neurais Artificiais: fundamentos e aplicações, um texto básico**. 4. ed. São Paulo: Livraria da Física, 2006.

- LI, J. et al. Building recognition in urban environments: A survey of state-of-the-art and future challenges, **Information Sciences**, v. 277, p. 406-420, 2014.
- LI, J., ALLINSON, N, M. Subspace learning-based dimensionality reduction in building recognition, **Neurocomputing**, v. 73, p. 324-330, 2009.
- LI, J.; ALLINSON, N. M. Building recognition using local oriented features, **IEEE Trans. Ind. Inform**, v. 9, n. 3, p. 1697-1704, 2013.
- LI, Y., SHAPIRO, L.G. Consistent line clusters for building recognition in CBIR, **Proc. IEEE Int'l. Conf. Pattern Recognition**, p. 952-956, 2002.
- LOWE, D.G. Distinctive image features from scale-invariant keypoints, **Int. J. Comput. Vision**, v. 60, n. 2, p. 91-110, 2004.
- MANZAN, J. R. G. **Avançada análise do uso de novos vetores-alvo em MLPs de alta performance**. Dissertação (Mestrado) – Universidade Federal de Uberlândia, Programa de Pós-Graduação em Engenharia Elétrica. Uberlândia – MG: 2012.
- MAYER, H. Automatic object extraction from aerial imagery – a survey focusing on buildings, **Comput. Vision Image Understanding**, v. 74, n. 2, p. 138-149, 1999.
- MISITI, M. et al. **Wavelet toolbox: user's guide**. Natick: Mat Works, 2000.
- OLIVEIRA, H. M. **Análise de sinais para engenheiros: uma abordagem via wavelets**. Rio de Janeiro: Brasport, 2007.

- PASQUALI, et al. Extração de Características Utilizando Filtro de Gabor Aplicado à Identificação de Defeitos no Couro Bovino. **Hifen, Uruguaiana**, v. 31, n. 59/60, 2007.
- PEDRINI, H.; SCWARTZ, W. R. **Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações**. São Paulo: Thomson Learning, 2008.
- PIPARSANIYAN, Y., SHARMA, V. K., MAHAPATRA, K. K. Robust Facial Expression Recognition using Gabor Feature and Bayesian Discriminating Classifier, **In: Communications and Signal Processing (ICCSP), International Conference on**, p. 538-54, 2014.
- SILVA, I. N., SPATTI, D. H., FLAUZINO, R. A. **Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas - Curso Prático**. São Paulo: Artliber, 2010.
- SILVA, L. A. et al. Aplicação da transformada Wavelet na Redução de Ruído em Sinais de Voz. In: **XII CEEL**, 2014.
- SOARES, W, C. **Um Método Não-Limiar para Redução de Ruído em Sinais de Voz no Domínio Wavelet**. Tese (Doutorado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Ilha Solteira – SP: 2009.
- STRUC, V., PAVESIC, N. Gabor-Based Kernel Partial-Least-Squares Discrimination Features for Face Recognition, **Informatica**, v. 20, n. 1, p. 115-138, 2009.
- TRINH, H. -H. et al. Facet-based multiple building analysis for robot intelligence, **J. Appl. Math. Computation**, v. 205, n. p. 537-549, 2008.

WEEKS, M. **Processamento Digital de Sinais Utilizando Matlab e Wavelets**. 2.ed. Rio de Janeiro: LTC, 2012.

ZHANG, L. et al. Discovering discriminative graphlets for aerial image categories recognition, **IEEE Trans. Image Process**, v. 22, n. 12, p. 5071-5084, 2013.

ZHANG, W., KOSECKÁ, J. Hierarchical building recognition, **Image Vision Comput**, v. 25, n. 5, p. 704-716, 2007.

<http://eeepro.shef.ac.uk/building/dataset.rar>. Acesso em: 30 de Mar. de 2014.

<http://www.vision.ee.ethz.ch/showroom/zubud/>. Acesso em: 15 de Fev. de 2015.