UNIVERSIDADE FEDERAL DE PELOTAS

Centro de Desenvolvimento Tecnológico

Programa de Pós-Graduação em Computação

Dissertação

Uma proposta de arquitetura de rede neural convolucional intervalar para o processamento de imagens intervalares

Ivana Patrícia Iahnke Steim

Pelotas, 2017.

IVANA PATRÍCIA IAHNKE STEIM

Uma proposta de arquitetura de rede neural convolucional intervalar para o processamento de imagens intervalares

Dissertação apresentada ao Programa de Pós-Graduação em Computação do Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas, como requisito parcial à obtenção do título de Mestre em Ciência da Computação

Orientador: Marilton Sanchotene de Aguiar

Coorientadora: Aline Brum Loreto

Pelotas, 2017

Ficha Catalográfica

S818p Steim, Ivana Patrícia Iahnke.
Uma proposta de arquitetura de rede neural convolucional intervalar para o processamento de imagens intervalares / por Ivana Patrícia Iahnke Steim. – 2017.
113 f. : il. color.

Orientador: Prof. Marilton Sanchotene de Aguiar.

Dissertação (mestrado) - Universidade Federal de Pelotas, Programa de Pós-Graduação em Computação, Mestrado Profissional em Educação e Tecnologia, Pelotas, 2017.

1. Redes neurais convolucionais - Computação. 2. Aritmética intervalar. 3. Processamento digital de imagens. 4. Imagens intervalares. I. Aguiar, Marilton Sanchotene de. II. Universidade Federal de Pelotas – UFPel. III. Título.

CDD 006.32

Catalogação na publicação:
Bibliotecária Glória Acosta Santos CRB 10/1859
Biblioteca IFSul - Câmpus Pelotas

Agradecimentos

A minha filha Aléxia, que muito me incentivou e ajudou.

Ao meu marido Cláudio pelo cuidado.

Aos meus pais, Nilson e Carmen pelo constante apoio.

A minha irmã Silvana pelo encorajamento.

Aos meus orientadores Aline e Marilton, por todo conhecimento compartilhado.

Ao colega Lucas Tortelli, pela assistência.

A Adriane Borda, pelo entendimento.

A UFPel, pelos recursos e

A Deus, por tudo.

Os que desprezam os pequenos acontecimentos nunca farão grandes descobertas. Pequenos momentos mudam grandes rotas. (Augusto Cury)

Resumo

STEIM, Ivana Patrícia Iahnke. **Uma proposta de arquitetura de rede neural convolucional intervalar para o processamento de imagens intervalares.** 2017. Dissertação (Mestrado em Ciências da Computação) – Programa de Pós-Graduação em Ciências da Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2017.

O objetivo geral deste trabalho é a proposta de uma extensão intervalar para redes neurais convolucionais e a análise de sua aplicação em imagens digitais intervalares no contexto de reconhecimento de padrões em imagens, visando alta exatidão e confiabilidade nos resultados. Este trabalho contém uma rede neural convolucional intervalar com propósito de controlar e automatizar a análise do erro numérico, onde as camadas que compõem a rede neural por convolução são representadas por operações equivalentes através de intervalos e tem-se por objetivo analisar se houve melhora na precisão e na classificação. Primeiro, as imagens tradicionais são transformadas em imagens intervalares, considerando a vizinhança de 4 e de 8 de seus pixels; após é observado o processamento pela rede dessas imagens quanto à exatidão e controle de erro; o terceiro passo é inserir o conceito de fatiamento da imagem intervalar à procura de uma melhoria na capacidade de classificação da rede, com isso são observados alguns casos e seu efeito na acurácia da rede; por fim, são introduzidas operações de Validação Cruzada e de Image Augmentation para confirmar overfiting e buscar um melhor desempenho da rede, respectivamente. Observou-se que o recurso de fatiamento, admissível às imagens intervalares, mostrou-se como melhor opção para um melhor desempenho de classificação da rede nas configurações atuais desta.

Palavras–chave: redes neurais convolucionais; aritmética intervalar; processamento digital de imagens; imagens intervalares.

Abstract

STEIM, Ivana Patrícia Iahnke. **A proposed convolutional neural network architecture for the processing of interval images** 2017. Dissertation (Master's Degree in Computer Science) - Programa de Pós-Graduação em Ciências da Computação, Centro de Desenvolvimento Tecnológico, Universidade Federal de Pelotas, Pelotas, 2017.

The general objective of this work is the proposal of an interval extension for convolutional neural networks and the analysis of their application in interval digital images in the context of pattern recognition in images, aiming for high accuracy and reliability in the results. In this work, we have an intervalal convolutional neural network with the purpose of controlling and automating the numerical error analysis, where the layers that compose the convolutional neural network are represented by equivalent operations through intervals. accuracy and classification. Traditional images are transformed into interval images, considering the neighborhood of 4 and 8 of their pixels. Then the network processing of these images is observed for the accuracy and error control. Then the concept of interval image slicing is inserted, looking for an improvement in the classification capacity of the network. Some cases and their effect on network accuracy are observed. Cross-Validation and Image Augmentation operations are still introduced to confirm overfiting and seek better network performance, respectively. It was observed that the allowable feature of slicing interval images was the best option for a better classification performance of the network in its current configurations.

Key words: convolutional neural networks; interval arithmetic; digital image processing; interval images.

Lista de Figuras

Lista de Tabelas

Lista de Siglas

CNN – Convolutional Neural Network

MLP – MultiLayer Perceptron

PNG – Portable Network Graphics

L-BFGS – Limited Broyden Fletcher Goldfard

Lista de símbolos

$\varphi(\upsilon)$ – função de ativação

ΩN - conjunto dos pixels intervalares

ωx – operação unária

A(i,j) - matriz intervalar

aij - pixel intervalar

D - diferença de duas matrizes

D(p,q) - distância entre pixels p e q

D4(p,q) - distância de manhattan

D8(p,q) - distância tabuleiro de xadrez

De(p,q) - distância euclidiana

diam(W) - diâmetro do intervalo W

dist - distância

dist(X,Y) - distância entre dois intervalos A e B

Dm - métrica Intervalar de Moore

Dqm - quasi métrica intervalar

F(X) - Função intervalar de X intervalar

Graf(f) - gráfico de uma função

I - imagem intervalar

Iij - matriz intervalar identidade

IN - conjunto dos intervalos naturais

IN - conjunto dos intervalos naturais

inf(A) - ínfimo da matriz intervalar

IR - conjunto de todos os intervalos reais

$\mathrm{mag}(\nabla f)$ – magnitude do vetor gradiente

med(A) - ponto médio do intervalo

Mij - matriz resultante da multiplicação entre matrizes

N - conjunto dos naturais

N - conjunto dos números naturais

O(i,j) - matriz intervalar nula

P - máscara para detecção de pontos isolados

Pij - produto do intervalo I pela matriz A

sup(A) - supremo da matriz intervalar

T - intersecção entre matrizes

U - união entre matrizes

X - um intervalo

X^{I}_{ij} – extensão intervalar

Lista de equações

1 $$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2},$$

2 $$D_4(p,q) = |x-s| + |y-t|$$

3 $$D_8(p,q) = \max(|x-s|, |y-t|)$$

4 $$f(x,y) = f_R(x,y) + f_G(x,y) + f_B(x,y)$$

5 $$s_k = T(r_k) = \sum_{j=0}^{k} \frac{nj}{n} = \sum_{j=0}^{k} p_r(r_j)$$

6 $$P = W_1 . Z_1 + W_2 . Z_2 + \cdots + W_9 . Z_9 = \sum_{i=1}^{9} W_i . Z_i$$

7 $$mag(\nabla f) = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\delta f}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta y}\right)^2}$$

8 $$\nabla^2 f(x,y) = f[x+1, y] + f[x-1, y] + f[x, y+1] + f[x, y-1] - 4f[x, y]$$

9 $$\Delta_G(F)_{(x)} = (F \oplus G)_{(x)} = \bigvee [F(x-u) + G(u)] \qquad u \in dom(G)$$

10 $$\varepsilon_G(F)_{(x)} = (F \ominus G)_{(x)} = \bigwedge [F(x-u) - \hat{G}(u)] \qquad u \in dom(G)$$

11 $$\varepsilon_G(F)_{(x)} = (F \ominus G)_{(x)} = \bigwedge [F(x+u) - G(u)] \qquad u \in dom(G)$$

12 $$u_k = \sum_{j=1}^{m} w_{kj} x_j$$

13 $$y_k = \varphi(u_k + b_k)$$

14 $$v_k = u_k + b_k$$

15 $$y_k = \frac{1}{1 + e^{(-v_k)}}$$

16 $$e_k(n) = d_k(n) - y_k(n)$$

17 $$E(n) = \frac{1}{2} \sum_{k \in C} e_k^2(n)$$

18 $$E_{med} = \frac{1}{N} \sum_{n=1}^{N} E(n)$$

19 $$p' = \left[p - \frac{d}{2}; p + \frac{d}{2}\right]$$

20 $$X_{ij}^{l} = \sum_{a}^{m-1} \sum_{b}^{m-1} w_{ab} \Im_{(i+a)(j+b)}^{l-1}$$

21 $$X_{ij} = \max \left\{ X_{i'j'} : i \le i' < i + k,\ j \le j' < j + k \right\}$$

22 $$x_{ij}^{l} = m\left(X_{ij}^{l}\right) = m\left(\left[\underline{x_{ij}^{l}}; \overline{x_{ij}^{l}}\right]\right) = \frac{\underline{x_{ij}^{l}} + \overline{x_{ij}^{l}}}{2}$$

23 $$Diâmetro(X) = w(X) = \overline{x} - \underline{x}$$

24 $$Erro\ Absoluto = |x - m(X)| < \frac{w(X)}{2}$$

25 $$Dp = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}}$$

26 $$Erro\ Relativo = \left|\frac{x - m(X)}{x}\right| \le \frac{w(X)}{2min|x|}\ se\ 0 \notin X$$

Sumário

1. INTRODUÇÃO

A precisão de cálculos dentro do processamento digital de imagens é muito importante para algumas aplicações. Este trabalho busca apresentar um estudo da aplicação de imagens intervalares em uma rede neural por convolução intervalar (Tortelli, 2016) no reconhecimento de padrões. O processamento de imagens intervalares, segundo Cruz et al. (2010), tem se mostrado uma ferramenta poderosa na análise e controle de erros no processamento de imagens tradicional.

Embora a aritmética intervalar e o processamento de imagens digitais já possuam quantidade razoável de bibliografia, o processamento digital de imagens intervalares pode ser considerado uma teoria recente, segundo Takahashi, Bedregal e Lyra (2004, 2005).

Este trabalho tem como finalidade a utilização de uma Rede Neural por Convolução Intervalar, com imagens intervalares conforme proposta de LYRA (2003), visando controlar a propagação do erro numérico gerado durante todo o processo e obter resultados com maior exatidão.

1.1. Motivação

A quantidade de áreas que utilizam o processamento de imagens digitais é muito vasta, desde campos que atuam em transformações na qualidade da imagem até setores de exames médicos, biometria ou mesmo visão computacional, permitindo a locomoção de robôs, a identificação de objetos.

O processamento de imagens também compreende uma área vasta, não existindo concordância entre autores de sua delimitação exata, referindo-se as áreas limites de análise de imagens ou visão computacional (GONZALEZ, 2010).

O Processamento de Imagens comporta diversas técnicas, para a extração de informações em uma imagem. Existem diversas operações morfológicas que atuam na descrição, análise ou alteração das formas contidas na imagem (CRUZ 2008).

A segmentação é uma etapa fundamental deste processamento, quando a imagem deve ser subdividida em áreas ou objetos significantes e qualquer erro nesta será propagado à análise de imagens, prejudicando seu desempenho (GONZALEZ, 2010; TAKAHASHI; BEDREGAL; LYRA, 2004, 2005)).

Atualmente a aritmética intervalar tem sido utilizada associada a outras áreas no intuito de dirimir o erro computacional ou melhor, “obter um controle rigoroso de diversos tipos de erros computacionais envolvendo representações finitas de

números reais" (TAKAHASHI; BEDREGAL; LYRA, 2005, p.2). E assim o processamento de imagens digitais foi interligado à matemática intervalar, quando Lyra estendeu a noção clássica de imagens digitais para o modelo de imagem intervalar digital, onde cada pixel representa a intensidade de maneira intervalar (BEDREGAL et al., 2003).

Segundo Russel; Norvig (2013), Redes Neurais Artificiais consistem em uma estrutura de aprendizado de máquina. Estas redes são especificadas a partir de um conjunto de neurônios de entrada, que recebem dados, valores numéricos ou mesmo imagens como é o caso das Redes Neurais Convolucionais (do inglês, *Convolutional Neural Network* – CNN). As informações a serem processadas são repassadas às camadas de funções que estimulam ou não estes neurônios. No contexto de processamento de imagens, as CNNs são uma classe de algoritmos de aprendizagem de máquina, capazes de aprender com as informações presentes na imagem através de convoluções.

Assim, a motivação deste trabalho vem da junção das áreas de Fundamentos da Computação com Sistemas Inteligentes, com o uso da aritmética intervalar em reconhecimento e processamento de imagens intervalares em uma rede neural simples.

1.2. Objetivos

Neste trabalho visa-se desenvolver a exploração das interações da matemática intervalar, em métodos de processamento de imagens, neste caso utilizando as redes neurais convolucionais. E também explorar o tipo de imagens que serão tratadas, como as imagens digitais intervalares, visando garantir altos níveis de exatidão e aprimorar a qualidade do reconhecimento de padrões em imagens na rede.

Um objetivo geral deste trabalho é a proposta de uso de imagens intervalares em uma rede neural por convolução intervalar e a análise de sua aplicação em imagens digitais intervalares no contexto de reconhecimento de padrões em imagens visando alta exatidão e confiabilidade nos resultados.

1.3. Organização deste trabalho

Este texto está dividido em 3 capítulos, sendo o segundo capítulo responsável pela apresentação da revisão bibliográfica, com sua primeira parte apresentando a aritmética intervalar, onde são abordados seus principais conceitos, definições e fundamentos, como propriedades algébricas, topologia e funções. A matemática intervalar é uma teoria que, em computação, serve para evitar erros de truncamento ou arredondamento que ocorrem ao discretizarmos valores contínuos, como ocorre ao representarmos a imagem em meio digital.

Na segunda parte da revisão bibliográfica tem-se o processamento de imagens digitais, com informações sobre o processamento exercido sobre a imagem com intuito de obter informações ou alterá-la. Discorre sobre conceitos de imagem, de pixels, de distâncias, discretização espacial e em amplitude, operações realizadas sobre os pixels, morfologia matemática, etapa de segmentação e aplicações deste processamento de imagens digitais.

A terceira parte da revisão apresenta o conceito de imagens digitais intervalares, unindo os conceitos da aritmética intervalar nas técnicas do processamento digital. Trás conceitos de trabalhos correlatos. Por imagem intervalar entende-se que cada pixel da imagem representa um intervalo que conterá o valor pontual da imagem original tradicional. A imagem pode ser vista como uma matriz destes pixels intervalares, e neste sub-capítulo serão re-expostos temas, sob a ótica intervalar, da morfologia matemática, operações sobre pixels e etapa de segmentação.

Na quarta parte deste capítulo de revisão bibliográfica abordam-se conceitos e noções gerais sobre redes neurais artificiais, é apresentada a definição de neurônio artificial, aprendizagem, algumas aplicações e certas arquiteturas para estas redes.

No terceiro capítulo são apresentadas as atividades desenvolvidas para a utilização de rede neural convolucional intervalar com imagens intervalares e os resultados obtidos em seus testes de utilização.

No quarto capítulo são expressas as considerações finais percebidas ao final deste estudo, com impressões absorvidas e sugestões para trabalhos futuros.

2. REVISÃO BIBLIOGRÁFICA

2.1. ARITMÉTICA INTERVALAR

A Matemática Intervalar é uma teoria proposta por Moore (1966) e que pode proporcionar uma solução ao controle rigoroso e automático de erros presentes em operações numéricas computacionais e no tratamento e modelagem da incerteza na computação. É um recurso na tentativa de minimizar erros provenientes de arredondamentos ou truncamentos numéricos, quando na transformação de valores contínuos a discretos, o que ocorre usualmente na obtenção de dados por meios analógicos ou de informações do mundo real e que precisam ser representados em meios computacionais.

Serão apresentados neste capítulo conceitos e definições pertinentes do conjunto de intervalos de números reais, como operações aritméticas, princípios de monotocidade e inclusão na aritmética intervalar. Tais conceitos e definições são fundamentais para o entendimento e aplicação da teoria intervalar no reconhecimento e processamento de imagens.

2.1.1. Fundamentos da Aritmética Intervalar

Como uma definição básica, tem-se que um Intervalo Real é o subconjunto não vazio dos números reais, fechado e limitado. Considerando que x_1 e x_2 pertençam ao conjunto dos números Reais, tal que x_1 seja menor ou igual a x_2, então o conjunto $\{x \epsilon R / x_1 \leq x \leq x_2\}$ é um intervalo real, denotado por $X = [x_1; x_2]$, onde x_1 é o limite inferior do intervalo e x_2 é o limite superior do mesmo. O conjunto IR é o conjunto de todos os intervalos reais, ou seja, $IR = \{[x_1, x_2] / x_1, x_2 \in R, x_1 \leq x_2\}$.

Os intervalos representados por único ponto, como $X = [x, x] \in R$ são chamados de intervalos degenerados ou intervalos pontuais (Oliveira et al., 1997).

A seguinte cadeia de inclusões entre conjuntos é válida: $N \subseteq Z \subseteq Q \subseteq R \subseteq IR$.

A igualdade entre dois intervalos é dada a partir da noção de igualdade entre conjuntos, sendo: $A = B \Leftrightarrow \forall x \in A \Rightarrow \forall x \in B$ e $\forall x \in B \Rightarrow x \in A$, ou seja, para todo número pertencente ao conjunto A existe um correspondente no conjunto B e vice-versa.

Neste caso, considerando os intervalos A=[a1, a2] e B=[b1, b2], pode-se afirmar que A = B se e somente se a1 = b1 e a2 = b2.

Obs.: Usualmente os intervalos são nomeados por letras em caixa maiúscula, e seus respectivos limites inferiores e superiores do intervalo utilizam a mesma letra, em caixa minúscula, como A = [a_1; a_2], F = [f_1; f_2], X = [x_1;x_2], ... e outra forma de representação seria $A=[\underline{a},\bar{a}]$

2.1.1.1. Operações Aritméticas

As operações aritméticas de soma, subtração, multiplicação e divisão dos conjuntos de Intervalos Reais são definidas por $A * B = \{a * b \ / \ a \in A \ e \ b \in B\}$ e $* \in \{+, -, \times, /\}$ ou seja, a regra é válida para as quatro operações aritméticas citadas. Na operação de divisão deve ser assumido que 0 $\notin$ B ou a operação estará mal definida. As definições apresentadas neste subitem foram baseadas em Oliveira et al. (1997) e Vargas (2006).

Considerando ω como uma operação unária, então ωX é definida por $\omega X = \omega(X) = \{\omega(x) \ / \ x \in X\} = [\min\{\omega(x) \ / \ \ x \in X\}; \max\{\omega(x) \ / \ x \ \in X\}]$.

A **soma intervalar** dos intervalos *X* e *Z* $\in$ IR, sendo $X = [x_1; x_2]$ *e* $Z = [z_1; z_2]$ é dada pela expressão: $X + Z = [(x_1 + z_1); (x_2 + z_2)]$. Esta provém da definição de intervalos e operações aritméticas intervalares, tendo-se que $x \in X \Rightarrow x_1 \leq x \leq x_2$ *e* $z \in Z \Rightarrow z_1 \leq z \leq z_2$, então $x_1 + z_1 \leq x + z \leq x_2 + z_2$.

O **pseudo inverso aditivo** de um intervalo *Y* $\in$ IR, sendo $Y = [y_1; y_2]$ é dado por:$-Y = [-y_2; -y_1]$. Esta deriva da definição de intervalos e operações aritméticas intervalares, onde $y \in Y \Rightarrow y_1 \leq y \leq y_2$. Multiplicando por *(-1)* tem-se: $-y_1 \geq -y \geq -y_2$, ou seja, $-y_2 \leq -y \leq -y_1$.

A **subtração intervalar**, considerando *X* e *Z* $\in$ IR, com $X = [x_1; x_2]$ *e* $Z = [z_1; z_2]$, é dada pela expressão: $X - Z = X + (-Z) = [(x_1 - z_2); (x_2 - z_1)]$. Esta pode ser deduzida a partir das definições e teoremas anteriores.

A **multiplicação intervalar**, considerando X e Z $\in$ IR, com $X = [x_1; x_2]$ *e* $Z = [z_1; z_2]$, é dada pela expressão:

$$X.Z = [\min\{x_1.z_1, x_1.z_2, x_2.z_1, x_2.z_2\}; \ max\{x_1.z_1, x_1.z_2, x_2.z_1, x_2.z_2\}].$$

A operação de multiplicação deve ser fechada em IR, então $a.b \in A.B$, sempre que $a \in A \, e \, b \in B$, ou seja, $A.B = [min\{a.b \,/\, a \in A, b \in B\}; max\{a.b \,/\, a \in A, b \in B\}]$.

O **pseudo inverso multiplicativo** intervalar é dado por: $Y^{-1} = \frac{1}{Y} = \left[\frac{1}{y_2}; \frac{1}{y_1}\right]$. Pela definição de intervalos e operações aritméticas intervalares tem-se: $y \in Y \Rightarrow y_1 \leq y \leq y_2$, portanto $\frac{1}{Y} \leq \frac{1}{y_1} \; e \; \frac{1}{y_2} \leq \frac{1}{y} \quad assim \quad \frac{1}{y_2} \leq \frac{1}{y} \leq \frac{1}{y_1}$.

Consequentemente $\frac{1}{y} \in \left[\frac{1}{y_2}; \frac{1}{y_1}\right] = \frac{1}{Y}$, sempre que $y \in Y$.

A **divisão intervalar** dos intervalos X e $Z \in$ IR, sendo $X = [x_1; x_2]$ e $Z = [z_1; z_2]$ e $0 \notin Z$, é dada pela expressão:

$$\frac{X}{Z} = X * Z^{-1} = \left[\min\left\{\frac{x_1}{z_2}, \frac{x_1}{z_1}, \frac{x_2}{z_2}, \frac{x_2}{z_1}\right\}; \max\left\{\frac{x_1}{z_2}, \frac{x_1}{z_1}, \frac{x_2}{z_2}, \frac{x_2}{z_1}\right\}\right].$$

Esta decorre das definições anteriores de multiplicação e pseudo inverso multiplicativo intervalares.

Oliveira et al. (1997) elaboraram uma tabela que otimiza os cálculos de multiplicação e de divisão, considerando os sinais dos valores dos extremos dos intervalos. Esta tabela seria útil na implementação em computadores destas operações. Têm-se nove casos elencados para cada operação, de multiplicação e divisão, respectivamente.

Tabela 1 – Otimização de operações aritméticas

1. Multiplicação	$x_1 \geq 0\ e\ z_1 \geq 0$	$\Rightarrow X.Z =$	$[x_1.z_1; x_2.z_2]$
2. Multiplicação	$x_1 \geq 0\ e\ z_1 > 0 \leq z_2$	$\Rightarrow X.Z =$	$[x_2.z_1; x_2.z_2]$
3. Multiplicação	$x_1 \geq 0\ e\ z_2 < 0$	$\Rightarrow X.Z =$	$[x_2.z_1; x_2.z_2]$
4. Multiplicação	$x_1 < 0 \leq x_2\ e\ z_1 \geq 0$	$\Rightarrow X.Z =$	$[x_1.z_2; x_2.z_2]$
5. Multiplicação	$x_1 < 0 \leq x_2\ e\ z_1 \leq 0 \leq z_2$	$\Rightarrow X.Z =$	$[\min\{x_1.z_2, x_2.z_1\}; \max\{x_1.z_1, x_2.z_2\}]$
6. Multiplicação	$x_1 < 0 \leq x_2\ e\ z_2 < 0$	$\Rightarrow X.Z =$	$[x_2.z_1; x_1.z_1]$
7. Multiplicação	$x_2 < 0\ e\ z_1 \geq 0$	$\Rightarrow X.Z =$	$[x_1.z_2; x_2.z_1]$
8. Multiplicação	$x_2 < 0\ e\ z_1 < 0 \leq z_2$	$\Rightarrow X.Z =$	$[x_1.z_2; x_1.z_1]$
9. Multiplicação	$x_2 < 0\ e\ z_2 < 0$	$\Rightarrow X.Z =$	$[x_2.z_2; x_1.z_1]$
1. Divisão	$x_1 > 0\ e\ z_1 > 0$	$\Rightarrow \frac{X}{Z} =$	$[x_1/z_2; x_2/z_1]$
2. Divisão	$x_1 > 0\ e\ 0 \in [z_1; z_2]$	$\Rightarrow \frac{X}{Z} =$	*Não definida*
3. Divisão	$x_1 > 0\ e\ z_2 < 0$	$\Rightarrow \frac{X}{Z} =$	$[x_2/z_2; x_1/z_1]$
4. Divisão	$x_1 < 0 < x_2\ e\ z_1 > 0$	$\Rightarrow \frac{X}{Z} =$	$[x_1/z_1; x_2/z_1]$
5. Divisão	$x_1 < 0 < x_2\ e\ 0 \in [z_1; z_2]$	$\Rightarrow \frac{X}{Z} =$	*Não definida*
6. Divisão	$x_1 < 0 < x_2\ e\ z_2 < 0$	$\Rightarrow \frac{X}{Z} =$	$[x_2/z_2; x_1/z_2]$
7. Divisão	$x_2 < 0\ e\ z_1 > 0$	$\Rightarrow \frac{X}{Z} =$	$[x_1/z_1; x_2/z_2]$
8. Divisão	$x_2 < 0\ e\ 0 \in [z_1; z_2]$	$\Rightarrow \frac{X}{Z} =$	*Não definida*
9. Divisão	$x_2 < 0\ e\ z_2 < 0$	$\Rightarrow \frac{X}{Z} =$	$[x_2/z_1; x_1/z_2]$

Fonte: Oliveira, 1997

2.1.2. *Propriedades Algébricas da Soma e da Multiplicação*

As propriedades algébricas **Fechamento**, **Associatividade**, **Comutatividade** e **Elemento Neutro** da soma e da multiplicação em IR, com os intervalos reais A, B, C ∈ IR, são expressas por: (TRINDADE, 2009).

Tabela 2: Propriedades Algébricas da Soma e Multiplicação

	SOMA	**MULTIPLICAÇÃO**
FECHAMENTO	$Se\ A\ \in IR\ e\ B\ \in IR\ então\ A+B\ \in IR$	$Se\ A\ \in IR\ e\ B\ \in IR\ então$ $A.B\ \in IR$
ASSOCIATIVIDADE	$A+(B+C)=(A+B)+C$	$A.(B.C)=(A.B).C$
COMUTATIVIDADE	$A+B=B+A$	$A.B=B.A$
ELEMENTO NEUTRO	$\exists!\ 0=[0;0]\in IR\ tal\ que$ $A+0=0+A=A$	$\exists!\ 1=[1;1]\in IR\ tal\ que$ $A.1=1.A=A$
SUBDISTRIBUTIVIDADE		$A.(B+C)\subseteq(A.B)+(A.C)$

Fonte:Trindade, 2009

Outras propriedades algébricas relevantes do conjunto *IR*, considerando *Y* ∈ *IR* (OLIVEIRA et al., 1997):

- O conjunto IR **não possui inverso aditivo**, ou seja, nem sempre existe um intervalo –Y, tal que *Y+(-Y) = 0*. Também **não possui inverso multiplicativo**, ou seja, nem sempre existe um intervalo Y^{-1} tal que *Y. (Y^{-1}) = 1*.

- **0 ∈ Y – Y**, podendo ser provado observando-se a sequência:

$Y-Y=[Y_1;Y_2]+[-Y_2;-Y_1]=[Y_1-Y_2;Y_2-Y_1]$. Como $y_1\leq y_2$ tem-se que $y_1-y_2\leq 0$ e $y_2-y_1\geq 0$, ou seja, $0\in Y-Y$.

- **1 ∈ Y/Y**, sendo que 0 ∉ Y. Pode ser provado de forma análoga pela sequência:

$$\frac{Y}{Y}=\frac{[y_1;y_2]}{[y_1;y_2]}=[y_1;y_2].\left[\frac{1}{y_2};\frac{1}{y_1}\right]=\left[\frac{y_1}{y_2};\frac{y_2}{y_1}\right]$$

Como $y_1\leq y_2$, segue que $\frac{y_1}{y_2}\leq 1$ e $\frac{y_2}{y_1}\geq 1$, ou seja, $1\in Y/Y$.

- O conjunto IR não tem divisores de zero, ou seja, se Y e W $\in$ IR e Y.W = 0 então Y = 0 ou W = 0. Então se obtivermos Y.W = [0;0] significa que Y = 0 ou W = 0, necessariamente.

2.1.3 Inclusão Monotônica

O teorema de **inclusão monotônica**, considerando A, B, C, D $\in$ IR, tais que A$\subseteq$C e B$\subseteq$D, apresenta as seguintes propriedades (SANTOS, 2001 e TRINDADE, 2009):

1. $A + B \subseteq C + D$
2. $-A \subseteq -C$
3. $A - B \subseteq C - D$
4. $A.B \subseteq C.D$
5. $1 / A \subseteq 1 / C$, sempre que $0 \notin C$
6. $A / B \subseteq C / D$, sempre que $0 \notin D$

Outras propriedades:

1. $A + B = A + C \Rightarrow B = C$
2. $B - A = C - A \Rightarrow B = C$
3. $A + B \subseteq A + C \Rightarrow B \subseteq C$
4. $A.0 = 0.A = [0; 0]$
5. $\forall \alpha, \beta \in$ R vale que $(\alpha.\beta).A = \alpha.(\beta.A)$
6. $\forall \alpha, \beta \in$ R vale que $(\alpha + \beta).A \subseteq (\alpha.A) + (\beta.A)$
7. $\forall \alpha \in$ R vale que $\alpha.(A + B) = (\alpha.A) + (\alpha.B)$

Um intervalo é **simétrico**, por definição, quando -A = A. Assim, todo intervalo simétrico é da forma: [-a; a], com $a \geq 0$. Sendo $A = [a_1; a_2]$ um intervalo, por hipótese considere-se que A é um intervalo simétrico, segue que -A = A. Daí tem-se que $-A = [-a_2; -a_1] = [a_1; a_2] = A \Leftrightarrow a_1 = -a_2$ e $a_2 = -a_1$. Estabelecendo-se que $a_2 = a$, tem-se que $a = -a_1$, $\therefore a_1 = -a$. Logo, A é simétrico se e somente se A = [-a; a], com $a \geq 0$. Alguns exemplos de intervalos simétricos: [-2; 2], [0; 0], [$-\pi$; π].

Por corolário, tem-se que, se X $\in$ IR e é um intervalo simétrico, então

X = |X| * [-1; 1]. Este corolário provém, de forma imediata, da definição de intervalo simétrico e das propriedades do módulo de intervalos.

Sendo A, X, Y ∈ IR intervalos de reais, com X e Y simétricos. As **propriedades de intervalos simétricos** valem:

1. $A + X = A - X$
2. $A.X = |A|.X$
3. $A.X = |A|.|X|.[-1; 1]$
4. $A.(X \pm Y) = (A.X) \pm (A.Y)$

2.1.4 Intersecção, União e União Convexa em IR

Outras operações que podem ser definidas sobre o conjunto dos Intervalos Reais são a Intersecção, a União e a União Convexa (MESQUITA, 2002).

Intersecção entre dois intervalos A e B, sendo $A = [a_1; a_2]$ *e* $B = [b_1; b_2]$, é definida por: $A \cap B = [\max\{a_1, b_1\}; \min\{a_2, b_2\}]$, se $\max\{a_1, b_1\} \leq min\{a_2, b_2\}$.

Se $min\{a_2, b_2\} < \max\{a_1, b_1\}$, então $A \cap B = \phi$.

Uma propriedade da intersecção, considerando que os intervalos A,B,C,D ∈ IR, é dada por: Se A ⊆ C e B ⊆ D, então, A∩B ⊆ C∩D.

União de dois intervalos A e B, sendo A = [a_1; a_2] e B = [b_1; b_2], e A∩B ≠ 0, é definida por: A ∪ B = [min{a_1, b_1} ; max {a_2, b_2}].

União Convexa entre dois intervalos quaisquer A = [a_1; a_2] e B = [b_1; b_2] é definida por $\overline{A \cup B}$ = [min {a_1; b_1}; max {a_2; b_2}]. A intersecção de A e B pode ser vazia, e neste caso, o intervalo resultante será o intervalo de menor diâmetro que contém simultaneamente ambos os intervalos da operação.

2.1.5 A Topologia de IR

As propriedades topológicas do espaço de IR estão baseadas principalmente nas noções de proximidade e limite, como por exemplo, a distância, e usualmente possuem interpretação geométrica intuitiva (OLIVEIRA et al., 1997; SANTOS, 2001).

2.1.5.1 Definições Básicas

- A distância entre dois intervalos X e Y, sendo $X = [x_1; x_2]$ *e* $Y = [y_1; y_2]$ dois intervalos de IR, é definida como sendo o número real não negativo $\delta = \max\{|x_1 - y_1|, |x_2 - y_2|\}$.

Notação:

$$dist(X, Y) = dist([x_1; x_2], [y_1; y_2]) = \max\{|x_1 - y_1|, |x_2 - y_2|\} \geq 0$$

Corolário:

$$X = Y \Leftrightarrow dist(X, Y) = 0$$

- IR é um espaço métrico completo, munido com a função dist (X, Y**)** **(**ALEFELD**,** 1983**)**.

- Métricas sobre IR da função distância:
  - dist (A, B) = 0 $\Leftrightarrow$ A = B, $\forall$ A, B $\in$ IR
  - dist (A, B) = dist (B, A), $\forall$ A, B $\in$ IR
  - dist (A, C) $\leq$dist (A, B) + dist (B, C), $\forall$ A, B, C $\in$ IR

Geometricamente a distância entre dois intervalos é o comprimento do maior segmento que separa os respectivos extremos dos intervalos.

- O Módulo de um intervalo, considerando A = [a1, a2] $\in$ IR um intervalo, é definido como sendo o número real não negativo μ = dist(A, 0), que corresponde à distância de A ao zero.
  - Notação: |A| = | [a1; a2] | = dist(A; 0) = max {| a1 |, | a2 |} $\geq$ 0
  - Propriedades do Módulo
    1. | X | = 0 $\Leftrightarrow$ X = 0
    2. | X + Y | $\leq$ | X | + | Y |
    3. | X . Y | $\leq$ | X | . | Y |

Geometricamente o módulo de um intervalo é o comprimento do maior segmento que une cada um dos extremos do intervalo à origem.

- Teorema 2.3 (em OLIVEIRA et al., 1997) Sejam A, B, C, D $\in$ IR Intervalos. Então valem:
  1. $\mathrm{dist}(\mathrm{A}+\mathrm{B}, \mathrm{A}+\mathrm{C}) = \mathrm{dist}(\mathrm{B}, \mathrm{C})$;
  2. $\mathrm{dist}(\mathrm{A}.\mathrm{B}, \mathrm{A}.\mathrm{C}) \leq |\mathrm{A}|.\mathrm{dist}(\mathrm{B}, \mathrm{C})$;
  3. $\mathrm{dist}(\mathrm{A}+\mathrm{B}, \mathrm{C}+\mathrm{D}) \leq \mathrm{dist}(\mathrm{A}, \mathrm{C}) + \mathrm{dist}(\mathrm{B}, \mathrm{D})$
  4. $\mathrm{dist}(\mathrm{A}.\mathrm{B}, \mathrm{C}.\mathrm{D}) \leq |\mathrm{B}|.\mathrm{dist}(\mathrm{A}, \mathrm{C}) + |\mathrm{C}|.\mathrm{dist}(\mathrm{B}, \mathrm{D})$;
  5. $\mathrm{dist}(\mathrm{A}, \mathrm{B}) \leq |\mathrm{A}|.|\mathrm{B}|.\mathrm{dist}\left(\frac{1}{\mathrm{A}}, \frac{1}{\mathrm{B}}\right), \text{se } 0 \notin \mathrm{A}, 0 \notin \mathrm{B}$;
  6. $\mathrm{A} \subseteq \mathrm{B} \Longrightarrow |\mathrm{A}| \leq |\mathrm{B}|$;
  7. $|\mathrm{X}^{\mathrm{n}}| = |\mathrm{X}|^{\mathrm{n}}$.

Prova: Estão provados os itens 4 e 5 em Oliveira et al. (1997, p. 27) conforme consta abaixo. E em Cláudio (1994, p.420) estão provados os itens

de 1 a 3 e o item 6. E em Alefeld (1983, p. 13) está provado o item 7 do teorema.

Prova item 4:

$$dist(A.B, C.D) \leq dist(A.B, B.C) + dist(B.C, C.D) =$$
$$dist(B.A, B.C) \leq |B|.dist(A, C) + |C|.dist(B, D);$$

- O Diâmetro de um intervalo Z $\in$ IR é definido como sendo o número real, não negativo, d = z2 – z1, considerando-se que Z = [z1; z2].
  - Notação: diam (Z) = diam ([z1; z2]) = z2 – z1 $\geq$0

Geometricamente o diâmetro de um intervalo é o comprimento do segmento que une os extremos do intervalo.

- O Ponto Médio de um Intervalo A = [a1; a2] $\in$ IR é definido como sendo o número real m = (a1 + a2) / 2

med (A) = med ([a1; a2]) = (a1 + a2) / 2

2.1.6 Funções Intervalares

Funções intervalares podem ser obtidas a partir de funções reais e sua definição formal é dada por:

Definição: Seja *f*: X $\rightarrow$ Y uma função. Se X = Dom (f) $\subseteq$ IR e Y = Cod (f) $\subseteq$ IR, X $\rightarrow$ *f* (X) então diz-se que *f* é uma função intervalar de uma variável intervalar.

Onde Dom (f) é o domínio da função e Cod (f) é o contra-domínio da função.

2.1.6.1 Definições básicas de funções:

- Produto Cartesiano

Sejam U e V conjuntos não vazios, U x V = {(u,v) | u $\in$ U, v $\in$ V}

- Funções

Sejam X e Y dois conjuntos não vazios. Uma função *f* de X em Y é um subconjunto do produto cartesiano X·Y definido por {(x, *f*(x)) | x $\in$ X, *f*(x) $\in$ Y}, de modo que para cada elemento x de X existe um único elemento y de Y tal que y = *f*(x).

Toda função *f*: X $\rightarrow$ Y constitui-se de três partes:

Domínio (X = Dom (f)), Contra-Domínio (Y = Cod(f)) e Regra da função (valor da função *f* no ponto x).

- Igualdade entre duas funções:

Considera-se que duas funções *f*: A→B e g: C→D são iguais se e somente se A = C e B = D e *f*(x) = *f*(g), para todo elemento x, ou seja, duas funções são iguais quando possuem o mesmo domínio, o mesmo contradomínio e a mesma lei de formação.

- Imagem de uma função, sendo *f*: X→Y, denominamos por imagem de X a função *f* do conjunto I = *f*(X) produzido pelos valores de y = *f* (x) que *f* adote em todos os pontos x de X. Assim:

I = *f* (X) = {y = *f*(x) ∈Y / x ∈ X} ⊆ Y

- Teoremas:
  - Seja *f*: X→Y uma função e sejam K, L subconjuntos de X. Então tem-se f (K ∪ L) = *f* (K) ∪ *f* (L)
  - Seja *f*: X→Y uma função e sejam K, L subconjuntos de X com K ⊆ L.

    Então: *f* (K) ⊆ *f* (L)

- O gráfico de uma função *f*, considerando que *f*: X→Y uma função, é o subconjunto Graf(f) do produto cartesiano X·Y, formado pelos pares ordenados (x,f(x)) para todo x ∈ X, ou seja:

Graf(f) = {(x, *f*(x)) ∈ X.Y / x ∈ X}

Exemplo:

Seja *f*: {2, 3, 7}→N X = {2, 3, 7}, tem-se x→*f*(x) = x^2 então

Graf(f) = {(2, 4), (3, 9), (7, 49)}

Obs.: Para que qualquer subconjunto G ⊆ X.Y represente um gráfico de uma função *f*: X→Y, é necessário e suficiente que, para cada x ∈ X exista um único ponto (x, y) ∈ G, cuja primeira coordenada seja x.

Com base na definição de igualdades entre funções pode-se deduzir que duas funções são iguais, se e somente se possuem o mesmo gráfico, ou seja, *f, g : X→Y são iguais se e somente se Graf(f) = Graf(g).*

2.1.6.2 Definição formal de Função Intervalar

Definição: Seja *f*: X → Y uma função. Se X = Dom(f) ⊆ IR e Y = Cod(f) ⊆ IR,

$X \rightarrow f(X)$ então é dito que *f* é uma **função intervalar** de uma variável intervalar.

2.1.7 Inclusão Intervalar

Dado $x \in R$, diz-se que $X \in IR$, ou seja, X pertence ao conjunto dos Intervalos Reais, e é uma inclusão intervalar de x, se $x \in X$.

2.1.8 Princípio da Monotocidade das Operações Intervalares

Sejam X, Y, Z, W $\in$ IR, intervalos tais que $X \subseteq Y$ e $Z \subseteq W$. Então vale que $X * Z \subseteq Y * W$ para quaisquer operações * $\in$ {+, -, ., /}, onde $0 \notin Z$ e $0 \notin W$ no caso da operação de divisão.

2.1.9 Imagem e Avaliação Intervalar

As funções intervalares podem ser compostas a partir de funções reais. Dispondo *f* como uma função real de variável real e W um intervalo tal que $W \subseteq$ Dom(f) e *f* é contínua em W. A imagem intervalar é definida da função *f* em W (ou imagem de *f* em W) como sendo o intervalo definido por: I = I(f, W) = [min {*f*(w) / w $\in$W}, max {f(w) / w $\in$W}].

- Observe-se que Y =*f*(w) = I(*f*, W), onde *f* é uma função real e W é um intervalo contido no domínio da função *f*.
- Observe-se que, se W = [w,w] é um intervalo pontual (degenerado), então Y = *f* (W) também é um intervalo pontual, expresso por: Y = [*f*(w), *f*(w)].

Portanto a função real está contida nesta extensão intervalar (OLIVEIRA, 1997).

Se W = [w1, w2] é um intervalo com diam(W) > 0, o que significa não ser um intervalo pontual ou degenerado, então I(*f*, W) é o intervalo de menor diâmetro que contém todos os valores reais de *f*(W), quando w $\in$ W.

Considerando que *f* é uma função real de variável real x e X é um intervalo, define-se **avaliação intervalar** de *f* em X (ou extensão intervalar de *f*) como sendo a função intervalar de F(X), definida de maneira em que cada ocorrência da variável real x é substituída pela variável intervalar X. Também as operações reais {+, -, $\times$, /} são permutadas pelas referentes operações intervalares, de forma que quando X = [x,x] for um intervalo pontual, então F(X) = *f*(x).

- Teoremas com propriedades básicas de I(*f*, X) e F(X):
  - Seja *f* uma função real de variável real e X,Y dois intervalos.

Se X ⊆ Y ⊆ Dom(*f*), então I (*f*, X) ⊆ Dom (*f*,Y) [CLAUDIO,1994]

- Seja *f* uma função real contínua de variável real x e F(X) a sua respectiva extensão intervalar. Então é válido inferir que I(*f*, X) ⊆ F(X) para todo X ⊆ Dom(*f*).

- Corolários
- Se X = [x,x], então I(*f*,X) = F(X)
- Se x ∈ X, então I(*f*,x) ∈ I (*f*,X) e f(x) ∈ F(X).

Portanto percebe-se que toda função real é obtida como o limite de sua correspondente extensão intervalar, quando lim diam(X) = 0.

Em resumo, $f(x) = lim_{X \to x}\, F(x)$

2.1.10 Matrizes e Vetores Intervalares

Uma matriz é dita intervalar quando cada elemento da matriz representa um intervalo. Ou seja, a matriz $A = (A_{ij})$ é uma matriz intervalar de ordem *m* por *n* se *A* for uma matriz com m linhas e n colunas, onde cada elemento A_{ij} é um intervalo (i representa a linha e j a coluna do elemento na matriz) (OLIVEIRA et al., 1997).

Exemplo de matriz de ordem 2x3:

$$A = \begin{pmatrix} [1;2] & [-3;4] & [0;0] \\ [-1;1] & [2;3] & [2;2] \end{pmatrix}$$

Um vetor intervalar representa uma linha ou coluna da matriz, quando esta possui m=1 ou n=1 respectivamente. Usualmente é chamado por vetor intervalar os vetores com apenas 1 coluna de elementos, ou seja, n=1. (vetor coluna)

2.1.10.1 Igualdade entre matrizes intervalares

Seja $A = (A_{ij})_{m \times n}$ e $B = (B_{ij})_{r \times s}$ considera-se que $A = B$ se, e somente se, r = m e s = n e $A_{ij} = B_{ij}$, para todos os índices $1 \leq i \leq n,\ 1 \leq j \leq m$, ou seja, A = B se e somente se, as matrizes possuírem a mesma ordem e todos os seus elementos correspondentes forem iguais (OLIVEIRA et al., 1997).

Exemplo de duas matrizes iguais:

$$A = \begin{pmatrix} [1,2] & [-1,1] \\ [1,e] & [0,0] \end{pmatrix} \qquad B = \begin{pmatrix} [\sqrt{1},\sqrt{4}] & [-1,1]^2 \\ exp[0,1] & [0,0] \end{pmatrix}$$

2.1.10.2 Matriz Intervalar Nula

Considera-se $O = (O_{ij})$ uma matriz intervalar nula, de ordem *m* por *n*, se todos os seus elementos forem nulos, ou seja, $(O_{ij}) = [0,0]$ para todos os índices de i, j (OLIVEIRA et al., 1997).

2.1.10.3 Matriz Intervalar Identidade

Diz-se que $I = (I_{ij})$ é a matriz intervalar identidade de ordem *m* por *n*, se todos os seus elementos da diagonal principal forem o intervalo identidade e se todos os seus demais elementos forem intervalos nulos, ou seja, $I_{ij} = [1,1]$ quando *i* = *j* e $I_{ij} = [0,0]$, se *i* ≠ *j* (OLIVEIRA *et al.*, 1997).

Exemplo de uma matriz identidade:

$$I = \begin{pmatrix} [1,1] & [0,0] & [0,0] \\ [0,0] & [1,1] & [0,0] \\ [0,0] & [0,0] & [1,1] \end{pmatrix}_{3\times3}$$

2.1.10.4 Operações com matrizes intervalares

Suas principais operações são a soma, produto por intervalo e a multiplicação entre matrizes (OLIVEIRA *et al.*, 1997).

- **Soma de matrizes intervalares**

Sejam $A = (A_{ij})$ e $B = (B_{ij})$ duas matrizes de mesma ordem, então a matriz S pode ser definida como a soma das matrizes A e B, ou seja, $S = A + B$ onde os elementos $S_{ij} = A_{ij} + B_{ij}$.

- **Produto de matriz por intervalo**

Seja $A = (A_{ij})$ uma matriz intervalar de ordem *m* x *n* e I um intervalo. O produto do intervalo I pela matriz A é a matriz $P = (P_{ij})$, onde cada elemento é dado por $P_{ij} = I \times A_{ij}$, para todos os índices *i,j* (OLIVEIRA *et al.*, 1997).

- **Diferença de matrizes intervalares**

Sejam $A = (A_{ij})$ e $B = (B_{ij})$ duas matrizes intervalares de mesma ordem, define-se a matriz diferença das matrizes A e B como sendo a matriz *D* = *A* – *B*, com elementos $D_{ij} = A_{ij} - B_{ij}$, para todos os índices *i, j.*

Por exemplo, considerando as matrizes:

$$A = \begin{pmatrix} [1;1] & [0;1] \\ [1;1] & [-1;1] \end{pmatrix} \text{ e } B = \begin{pmatrix} [-1;2] & [3;4] \\ [2;2] & [-6;-4] \end{pmatrix}$$

tem-se $D = A - B = \begin{pmatrix} [1;1] - [-1;2] & [0;1] - [3;4] \\ [1;1] - [2;2] & [-1;1] - [-6;-4] \end{pmatrix}$ lembrando que: **X - Z** = X + (- Z) **= [(x1 – z2) ; (x2 – z1)],** como já visto.

$$= \begin{pmatrix} [(1-2);(1-(-1)] & [(0-4);(1-3)] \\ [(1-2);(1-2)] & [(-1-(-4);(1-(-6)] \end{pmatrix} =$$

$$\begin{pmatrix} [-1;2] & [-4;-2] \\ [-1;-1] & [3;7] \end{pmatrix}$$

Obs.: Em Oliveira (1997) consta que é possível a obtenção da subtração de matrizes a partir da operação de soma e de um produto por intervalo, como:

$A - B = A + [-1; -1] \cdot B$

- **Multiplicação entre matrizes intervalares**

Seja $A = (A_{ij})$ uma matriz intervalar de ordem *m* por *p*, e $B = (B_{ij})$ uma matriz intervalar de ordem *p* por *n*. A multiplicação da matriz A pela matriz B é uma matriz intervalar $M = (M_{ij}) = A \times B$ de ordem *m* por *n*, cujos elementos são dados por $M_{ij} = \sum_{k=1}^{p} A_{ik} \times B_{kj}$

Exemplo de multiplicação entre matrizes intervalares:

Considerando as matrizes intervalares $A = \begin{pmatrix} [1;10] & [0;1] \\ [1;1] & [-1;1] \end{pmatrix}$

e $B = \begin{pmatrix} [-1;2] & [3;4] \\ [2;2] & [-6;-4] \end{pmatrix}$,

lembrando que *X.Z = [min{x1.z1, x1.z2, x2.z1, x2.z2}; max{x1.z1, x1.z2, x2.z1, x2.z2}]*, tem-se que $M = A \cdot B = \left(\left(\sum_{k=1}^{2} A_{ik} \cdot B_{kj} \right)_{ij} \right) =$

$$\begin{pmatrix} [1;10] \cdot [-1;2] + [0;1] \cdot [2;2] & [1;10] \cdot [3;4] + [0;1] \cdot [-6;-4] \\ [1;1] \cdot [-1;2] + [-1;1] \cdot [2;2] & [1;1] \cdot [3;4] + [-1;1] \cdot [-6;-4] \end{pmatrix} =$$

então: $= \begin{pmatrix} [-10;20] + [0;2] & [3;40] + [-6;0] \\ [-1;2] + [-2;2] & [3;4] + [-6;6] \end{pmatrix} =$

Considerando que $A + B = [(a1 + b1); (a2 + b2)]$,

tem-se: $= \begin{pmatrix} [-10;22] & [-3;40] \\ [-3;4] & [-3;10] \end{pmatrix}$

- **Intersecção de matrizes**

Sejam $A = (A_{ij})$ e $B = (B_{ij})$ duas matrizes intervalares de mesma ordem. A intersecção da matriz A com a matriz B é a matriz $T = A \cap B$, cujos elementos são $T_{ij} = A_{ij} \cap B_{ij}$, para todos os índices *i, j.* Caso exista um par de índices *i, j* em que $A_{ij} \cap B_{ij} = \emptyset$, então não existirá a intersecção das matrizes.

- **União de matrizes**

Sejam $A = (A_{ij})$ e $B = (B_{ij})$ duas matrizes intervalares de mesma ordem. A união da matriz A com a matriz B é a matriz $U = A \cup B$, cujos elementos são $U_{ij} = A_{ij} \cup B_{ij}$, para todos os índices *i, j.* Caso exista um par de índices *i, j* em que $A_{ij} \cap B_{ij} = \emptyset$, então não existirá a união das matrizes.

- **Inclusão de matrizes intervalares**

Sejam A e B duas matrizes intervalares de mesma ordem. Considera-se que a matriz *B* é uma inclusão para a matriz *A*, se todo elemento da matriz *A* está contido no seu respectivo índice da matriz *B*, ou seja, $A \subseteq B \Leftrightarrow A_{ij} \subseteq B_{ij}$, para todos os índices *i, j.* Verifica-se a validade da afirmação $A \subseteq B \Leftrightarrow A \cap B = A$.

2.1.10.5 Propriedades das matrizes intervalares

As propriedades básicas de operações matriciais intervalares podem ser resumidas pelo teorema obtido em Oliveira (1997) na página 77.

Sejam A, B, C matrizes intervalares de mesma ordem. Então são válidas:

1. $A + (B + C) = (A + B) + C$ (Associatividade)
2. $A + O = O + A = A$ (Elemento neutro da adição: matriz nula)
3. $A + B = B + A$ (Comutatividade)
4. $A \times I = I \times A = A$ *(Elemento neutro da multiplicação: matriz identidade)*
5. $(A + B) \times C \subseteq A \times C + B \times C$ (Inclusão da distributividade)

A prova deste teorema pode ser vista em Santos (2001, p. 77).

A lei associativa do produto não é válida para matrizes intervalares, ou seja,

$$A \times (B \times C) \neq (A \times B) \times C$$

2.1.10.6 Operações entre matrizes reais e intervalares

É possível a transformação das informações, contidas em matrizes reais, serem transpostas para matrizes intervalares e de forma recíproca. As definições a seguir referem-se a estas transferências.

As matrizes reais são transformadas naturalmente em matrizes intervalares pontuais, no entanto, a transferência de dados de matrizes intervalares para matrizes reais apresenta mais opções, como encontra-se em Oliveira *et al.* (1997) e Santos (2001):

- **Matriz diâmetro**

Seja $A = (A_{ij})$ uma matriz intervalar. A matriz diâmetro da matriz *A* é a matriz **real** onde cada elemento corresponde ao diâmetro do respectivo intervalo da matriz intervalar, ou seja, $diam(A) = (diam(A_{ij}))$.

Exemplo: $diam\begin{pmatrix}[1;1] & [2;4]\\ [2;6] & [-2;1]\end{pmatrix} = \begin{pmatrix}0 & 2\\ 4 & 3\end{pmatrix}$

- **Matriz ponto médio**

Seja $A = (A_{ij})$ uma matriz intervalar. A matriz ponto médio da matriz *A* é a matriz **real** onde cada elemento corresponde ao ponto médio do respectivo intervalo da matriz intervalar, ou seja, $med(A) = (\,med\,(A_{ij}))$

Exemplo: $med\begin{pmatrix}[1;1] & [1;3]\\ [2;10] & [-3;9]\end{pmatrix} = \begin{pmatrix}1 & 2\\ 6 & 3\end{pmatrix}$

- **Matriz Módulo**

Seja $A = (A_{ij})$ uma matriz intervalar. A matriz módulo da matriz *A* é a matriz **real** onde cada elemento corresponde ao módulo do respectivo intervalo da matriz intervalar, ou seja, $|A| = (|A_{ij}|)$.

Exemplo: $\begin{vmatrix}[1;1] & [-5;3]\\ [3;6] & [2;3]\end{vmatrix} = \begin{pmatrix}1 & 5\\ 6 & 3\end{pmatrix}$

- **Ínfimo da matriz intervalar**

Seja $A = (A_{ij})$ uma matriz intervalar. O ínfimo da matriz intervalar *A* é a matriz **real** cujos elementos são os extremos inferiores dos intervalos correspondentes da matriz intervalar *A*, ou seja, $\inf(A) = (\inf(A_{ij}))$.

Exemplo: $\inf\begin{pmatrix}[1;1] & [-2;4]\\ [2;3] & [-1;1]\end{pmatrix} = \begin{pmatrix}1 & -2\\ 2 & -1\end{pmatrix}$

- **Supremo da matriz intervalar**

Seja $A = (A_{ij})$ uma matriz intervalar. O supremo da matriz intervalar A é a matriz **real** cujos elementos são os extremos superiores dos intervalos correspondentes da matriz intervalar *A*, ou seja, $\sup(A) = (\sup(A_{ij}))$

Exemplo: $\sup\begin{pmatrix}[1;1] & [0;4]\\ [2;7] & [-2;1]\end{pmatrix} = \begin{pmatrix}1 & 4\\ 7 & 1\end{pmatrix}$

2.1.10.7 Distância entre matrizes intervalares

Sejam *A* e *B* duas matrizes intervalares de mesma ordem. A distância entre as matrizes intervalares *A* e *B* é a matriz real cujos elementos correspondem a distância entre os respectivos intervalos das matrizes *A* e *B*, ou seja, $dist(A,B) = (dist(A_{ij}, B_{ij}))$, para todos os índices *i,j.*

Exemplo: Sejam $A = \begin{pmatrix}[1;1] & [0;1]\\ [1;1] & [-1;1]\end{pmatrix}$ e $B = \begin{pmatrix}[-1;2] & [3;4]\\ [2;2] & [-6;-4]\end{pmatrix}$

$$dist(A,B) = \begin{pmatrix}dist([1;1],[-1;2]) & dist([0;1],[3;4])\\ dist([1;1],[2;2]) & dist([-1;1],[-6;-4])\end{pmatrix} = \begin{pmatrix}2 & 3\\ 1 & 5\end{pmatrix}$$

O presente capítulo teve como objetivo descrever as principais operações e propriedades da matemática intervalar, as quais são utilizadas para a obtenção dos resultados, conforme os objetivos.

2.2 PROCESSAMENTO DE IMAGENS DIGITAIS

Entende-se por processamento de imagens digitais ao tratamento exercido sobre a imagem, com intuito de obter informações desta ou realizar alterações sobre esta. Uma imagem pode ser obtida ou gerada por diversos meios, como câmeras de fotografia ou vídeo, aparelhos de ultrassom, microscópio eletrônico e computadores.

Segundo Gonzalez (2007), uma imagem pode ser definida como uma função bidimensional e(x,y), onde x e y são coordenadas espaciais no plano, e a amplitude de e, de qualquer par de coordenadas (x,y), é chamada de intensidade ou nível de cinza da imagem naquele ponto específico. Quando x, y e o valor de intensidade de e são todos finitos, ou seja, quantidades discretas, então esta imagem é digital.

Uma imagem digital é composta por um número finito de elementos e cada um possui uma localização e valor próprio. Estes são conhecidos por *pixels* ou *Picture elements* ou *image elements* ou *pels* (GONZALEZ, 2007).

O processamento digital de imagens engloba uma longa variedade de campos de aplicação, como no processamento e interpretação de imagens captadas por satélites, que podem ser usadas em geoprocessamento, meteorologia, geografia, entre outros. Na medicina, utilizam-se imagens para diagnóstico, oriundas, por exemplo, de Raio X (MARQUES FILHO, 1999). Também pode-se considerar o uso em detecções de faces ou objetos, na biometria, entre várias outras possíveis aplicações passíveis ao processamento de imagens digitais.

Não existe uma concordância entre autores a respeito de onde termina a área de processamento de dados e começam outras áreas, como análise de imagens e visão computacional. Há uma corrente que afirma que o processamento de imagens pertenceria a área em que tanto a entrada como a saída do processo seriam imagens. Gonzalez (2007) considera esta teoria limitante e de fronteira artificial, pois assim até uma tarefa trivial de computar a intensidade média da imagem poderia não ser considerada uma operação de processamento de imagens. Este autor, considerado como referência, adiciona ao conceito de abrangência do processamento de imagens digitais, os conjuntos de processos que extraem atributos de imagens, incluindo o reconhecimento de objetos individuais destas.

Outro paradigma usual para a identificação das fronteiras no processamento de imagens digitais, segundo Gonzalez (2007), é considerá-lo em três tipos de processos computacionais: baixo, médio e alto nível de processamento.

O baixo nível constitui-se por todos os processamentos em que tanto a entrada como a saída seriam compostos por imagens, como as operações primitivas de pré-processamento para reduzir ruídos, melhoria de contraste e nitidez da imagem.

O nível de médio de processamento é caracterizado pelas entradas serem geralmente imagens e as saídas serem atributos extraídos destas imagens, como bordas, contornos, e a identidade de objetos individuais presentes na imagem. Também composto por tarefas como segmentação e classificação de objetos.

O nível alto envolve dar significado a um conjunto de objetos reconhecidos, como análise de imagens e, por fim, executando as funções cognitivas normalmente associadas com a visão. Envolveria tarefas como análise de imagens e possível desenvolvimento de funções cognitivas normalmente associadas com a visão (GONZALEZ, 2007).

Uma sobreposição entre processamento de imagens digitais e análise de imagens é a área de reconhecimento de regiões ou objetos individuais na imagem.

Um exemplo fornecido por Gonzalez (2007) para uma melhor visualização dos limites do escopo de processamento de imagens foi uma análise automatizada de textos, onde uma imagem de texto é inserida para processamento, então é realizado um pré-processamento da imagem, extraindo por segmentação os caracteres individualmente, descrevendo estes adequadamente para processamento computacional e seu reconhecimento.

A aquisição da imagem, o aperfeiçoamento ou pré-processamento desta, a sua restauração, seu processamento de cor, sua compressão, seu processamento morfológico e a sua segmentação são alguns dos passos ditos como fundamentais ao processamento de imagens por Gonzalez (2007).

A aquisição da imagem pode ser feita através de sensores que possuem como saída uma forma de onda contínua, o que traria a necessidade de processamento para sua representação digital. O processo de discretização ou amostragem digitaliza as coordenadas desta imagem e o processo de quantização digitaliza os valores de amplitude ou intensidade do sinal, com representação em forma de matriz.

A próxima etapa de pré-processamento ou aperfeiçoamento da imagem apresenta diversos métodos para aprimorar a qualidade desta imagem (do inglês, *enhancement*). Pode empregar técnicas para diminuição de ruídos, para correções necessárias à aplicação futura da imagem, como utilizar filtros e outras operações que atuam sobre os pixels da imagem, transformando-a para adequar-se ao objetivo desejado.

A restauração da imagem também aborda a melhoria da aparência da imagem, porém se baseia em modelos matemáticos ou probabilísticos de degradação da imagem, sendo considerada uma etapa objetiva por Gonzalez (2007). Também é integrante do pré-processamento de imagem.

2.2.1 Representação formal da imagem

O pixel consiste na menor unidade de informação presente em uma imagem e a organização da imagem em forma de matriz de pixels geralmente possui simetria quadrada.

Cada pixel representado não possui as mesmas propriedades em todas as direções, sendo dito anisotrópico (DE ALBUQUERQUE, 2000). Ele possui quatro vizinhos de borda e outros quatro na diagonal. No processamento da imagem é necessário definir como será tratada a conectividade entre os pixels, considerando apenas os vizinhos de borda (B_4) ou considerando também os vizinhos da diagonal (B_8).

Outra questão a ser considerada é a distância entre um determinado ponto e seus vizinhos, visto que a distância entre pixels vizinhos de borda é correspondente a 1, e entre vizinhos diagonais é $\sqrt{2}$ (ESQUEF, 2002).

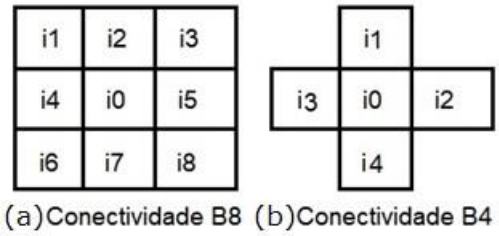


Figura 1 – Vizinhança de pixel
Fonte: Esquef, 2002

Uma imagem contínua pode ser representada computacionalmente por matrizes ou vetores contendo a posição (x,y) e o valor f(x,y) de cada pixel de uma

imagem digital, formando um arranjo bidimensional de pontos (DE QUEIROZ; GOMES, 2006).

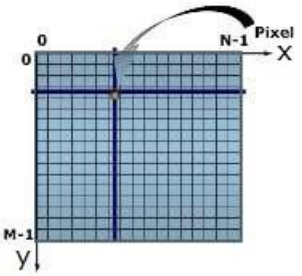


Figura 2 – Representação do sistema de coordenadas de uma imagem digital bidimensional
Fonte: De Queiroz; Gomes, 2006

As coordenadas dos vizinhos do pixel "i_0", considerando a conectividade B_4, podem ser dadas por (x+1, y), (x-1, y),(x, y+1)e(x, y-1). As coordenadas dos vizinhos do pixel "i_0", considerando a conectividade B_8, podem ser dadas por (x+1, y), (x-1, y),(x, y+1),(x, y-1), (x+1, y+1), (x+1, y-1),(x-1, y+1) e(x-1, y-1).

O conceito de distância pode ser relacionado ao conceito de conectividade, onde D_4 expressa a distância entre dois pixels conectados B_4. Considerando três pixels *p, q* e *z,* de coordenadas (x,y), (s,t) e (u,v) respectivamente, define-se a *Função Distância D*, com as propriedades (GONZALEZ 2007, MARQUES FILHO 1999):

- $D(p,q) \geq 0$ $(D(p,q) = 0 \text{ se e somente se } p = q)$
- $D(p,q) = D(q,p)$
- $D(p,z) \leq D(p,q) + D(q,z)$

A seguir serão apresentadas as definições de algumas das distâncias mais utilizadas, como as distâncias Euclidiana, Manhattan e de tabuleiro.

Distância Euclidiana (Distância Radial)

Para esta medida de distância, os pixels com distância euclidiana em relação a (x,y) menor ou igual a algum valor r, são os pontos contidos em um círculo de raio r centrado em (x,y) (MARQUES FILHO, 1999, p. 27), dada pela equação 1.

$$D_e(p,q) = \sqrt{(x-s)^2 + (y-t)^2}, \text{ onde } p = (x, y) \text{ e } q = (s, t). \quad (1)$$

$\sqrt{12}$	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{12}$
$\sqrt{6}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{6}$
2	1	0	1	2
$\sqrt{6}$	$\sqrt{2}$	1	$\sqrt{2}$	$\sqrt{6}$
$\sqrt{12}$	$\sqrt{6}$	2	$\sqrt{6}$	$\sqrt{12}$

Figura 3 – Ilustração de distância Euclidiana entre pixels
Fonte: Foresti, 2006.

Distância de Manhattan (**Distância D_4**)

Neste caso, os pixels tendo uma distância D_4 em relação a (x,y) menor ou igual a algum valor r formam um losango centrado em (x,y). Os pixels com $D_4 = 1$ são os 4-vizinhos de (x,y) (MARQUES FILHO, 1999, p.27), dada pela equação 2.

$$D_4(p,q) = |x-s| + |y-t| \quad (2)$$

Observando que nesta, o movimento ocorre apenas em sentido vertical ou horizontal, pois são considerados apenas os 4 vizinhos de bordas.

		2		
	2	1	2	
2	1	0	1	2
	2	1	2	
		2		

Figura 4 – Ilustração de distância de Manhattan entre pixels
Fonte: Foresti, 2006.

Distância de Tabuleiro de Xadrez (**Distância D_8**)

Neste caso, os pixels com distância D_8 em relação a (x,y) menor ou igual a algum valor formam um quadrado centrado em (x,y). Os pixels com $D_8 = 1$ são os 8-vizinhos de (x,y) (MARQUES FILHO, 1999, p.27), dada pela equação 3.

$$D_8(p,q) = \max(|x-s|, |y-t|) \quad (3)$$

onde usa-se a maior distância, pelo cálculo de $|x - s|$ ou $|y - t|$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Figura 5 – Ilustração de distância de Tabuleiro de Xadrez entre pixels
Fonte: Foresti, 2006.

A intensidade luminosa no ponto (x,y), sendo que x referencia a linha e y referencia a coluna, pode ser decomposta em: componente de iluminação e de reflectância, sendo f(x,y) = i(x,y) * r(x,y).

Em uma imagem colorida do sistema RGB, cada pixel pode ser visto como um vetor cujas componentes representam as intensidades de cores vermelho, verde e azul que compõem a sua cor. Então a imagem colorida pode ser vista como a composição de três imagens monocromáticas, ou seja, equação 4:

$$f(x,y) = f_R(x,y) + f_G(x,y) + f_B(x,y) \quad (4)$$

onde $f_R(x,y), f_G(x,y), f_B(x,y)$ representam, respectivamente, as intensidades luminosas das componentes vermelha, verde e azul no ponto (x, y).

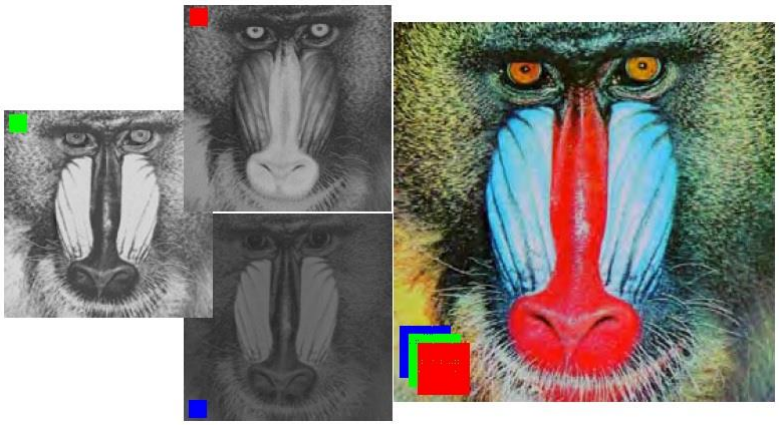


Figura 6 – Representação da composição de três planos monocromáticos.
Fonte: De Queiroz; Gomes, 2006.

2.2.2 Amostragem e Quantização

O processo de discretização espacial é chamado de amostragem, e o processo de discretização em amplitude é chamado por quantização. De forma básica, pode ser dito que a amostragem converte a imagem contínua (analógica) em uma matriz de M por N pontos ou pixels. M indica as linhas e N indica as colunas. Valores maiores de M e N implicam em uma imagem de maior resolução (MARQUES FILHO, 1999).

$$f(x,y) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{bmatrix}$$

A quantização faz com que cada um destes pixels assuma um valor inteiro, na faixa de 0 a $2^n - 1$. Quanto maior o valor de n, maior o nível de cinzas presentes na imagem digitalizada. Em geral, 64 níveis de cinza são considerados suficientes para o olho humano, mas, apesar disto, a maioria dos sistemas de visão artificial utiliza imagens com 256 níveis de cinza.

Considerando que cada elemento é uma aproximação do nível de cinza da imagem no ponto amostrado para um valor no conjunto {0, 1, ..., L-1}, costuma-se associar o nível mais baixo (0) com preto e o nível mais alto (L-1) com branco.

Para n=8, tem-se que L= 256, onde os tons de cinza poderão ser representados de 0 a 255, sendo a cor preta representada pelo valor 0 e a cor branca pelo valor 255.

Neste caso, o número de bits necessários para representar uma imagem digital será M x N x n. Ou seja, para ser utilizada uma matriz de 1024 linhas, 1024 colunas com n=8, ou, com 256 tons de cinza, precisa-se de 8.388.608 bits, ou 8Mb. (Precisamente $2^{10} * 2^{10} * 2^3 = 2^{23}$)

No caso de imagens coloridas o cálculo do número de bits necessários para representar uma imagem digital será M x N x n_R x n_G x n_B. Onde n_R, n_G e n_B representam os bits necessários à representação dos planos RGB, para as cores vermelho, verde e azul, respectivamente (DE QUEIROZ; GOMES, 2006).

A Tabela 3 contém o número de bytes empregados na representação de uma imagem digital monocromática para alguns valores típicos de M e N, com 2, 32 e 256 níveis de cinza.

Tabela 3 – Número de Bytes para uma imagem monocromática

M	**N**	**Número de Bytes (L)**		
		L = 2	L = 32	L = 256
480	640	38400	192000	307200
600	800	60000	300000	480000
768	1024	98304	491520	786432
1200	1600	240000	1200000	1920000

Fonte: De Queiroz; Gomes, 2006.

O número de amostras ou de níveis de cinza depende de características da imagem, como o seu tamanho (dimensão) ou complexidade, e também da aplicação a qual a imagem se destina. De Queiroz e Gomes (2006) ilustra a influência dos parâmetros de digitalização na qualidade visual de uma imagem digital monocromática com a Figura 7.

Figura 7– Influência da variação do número de amostras e de níveis de quantização na qualidade de uma imagem digital. (A) 200 x 200 pixels / 256 níveis; (B) 100 x 100 pixels / 256 níveis; (C) 25 x 25 pixels / 256 níveis; (D) 200 x 200 pixels / 2 níveis;
Fonte: De Queiroz; Gomes, 2006.

Os processos de amostragem e quantização podem ser aprimorados, buscando aprimorar a qualidade de imagem e diminuir a quantidade de bytes para armazenamento, utilizando técnicas adaptativas. Pensando sob a ótica da amostragem, a ideia seria utilizar um número maior de pontos em regiões da imagem com muito detalhamento, em detrimento de grandes regiões homogêneas. Sob a ótica da quantização, utilizar poucos níveis de cinza próximos a regiões onde ocorrem mudanças abruptas de intensidade, visto o olho humano não ser capaz de reconhecê-los (MARQUES FILHO, 1999).

A principal dificuldade na implementação de técnicas adaptativas para amostragem consiste na necessidade prévia de identificação das regiões e fronteiras presentes na imagem.

2.2.3 Operações Lógico Aritméticas sobre pixels da imagem

Após uma imagem ter sido adquirida e digitalizada, ela pode ser vista e manipulada como uma matriz de inteiros, através de operações lógicas e/ou aritméticas. Estas operações podem ser feitas sobre pixels individuais ou sobre um conjunto de pixels (vizinhança) (MARQUES FILHO, 1999).

Conforme Gonzalez (2007), operações sobre pixels são extensivamente usadas na maioria dos ramos de processamento de imagens. Denotam-se as operações aritméticas sobre dois pixels *p* e *q* como:

Tabela 4 – Operações aritméticas sobre pixels

Operações		**Uso**
Adição:	$p + q$	Principal uso na média de imagens para redução de ruídos
Subtração:	$p - q$	Útil na remoção de informação estática de fundo em imagens médicas
Multiplicação:	$p * q$ (ou pq ou p.q)	Seu principal uso é na correção de sombras de níveis de cinza
Divisão:	$p \div q$	

Fonte: Autor, baseada em texto de Gonzalez, 2007.

Operações aritméticas sobre imagens inteiras são realizadas pixel a pixel (GONZALEZ, 2007). Em contraste, as operações lógicas operam sobre regiões. Considerando as regiões *a* e *b,* as operações lógicas entre elas são:

Tabela 5 – Operações lógicas sobre regiões de pixels

E	Corresponde $a \cap b$
OU	Corresponde $a \cup b$
NÃO *a*	Corresponde ao complemento de *a*
XOU	Corresponde $a \cup b$ - $a \cap b$

Fonte: Inspirada em Foresti 2006

A Figura 8 ilustra as operações lógicas sobre regiões A e B.

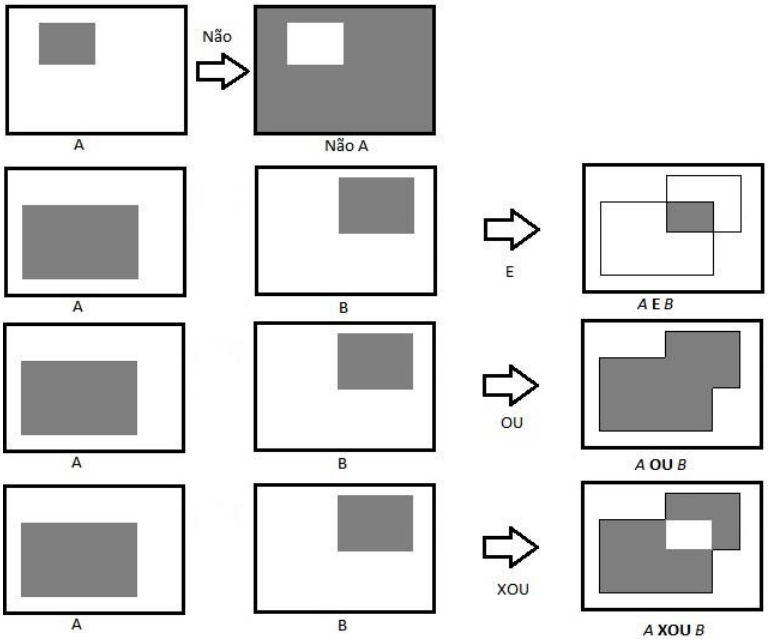


Figura 8 – Operações lógicas sobre regiões A e B
Fonte: Inspirada em Gonzalez, 2007.

2.2.4 Operações sobre Imagens

As técnicas de pré-processamento que visam aprimorar a imagem operam no domínio espacial ou no domínio de frequência. As que são no domínio espacial, baseiam-se em filtros que manipulam o plano da imagem, enquanto as que são de domínio da frequência atuam sobre o espectro da imagem. É comum misturar vários destes métodos para realçar determinadas características da imagem (ESQUEF, 2003).

As técnicas de realce são dependentes da aplicação, o que indica não haver uma técnica ótima para todos os casos. As técnicas de domínio espacial propõem a manipulação direta dos pixels da imagem, normalmente utilizando operações de convolução com máscaras, enquanto as técnicas de domínio de frequência propõem

modificações da Transformada de Fourier sobre a imagem (FORESTI, 2006), entre outras técnicas de domínio de frequência.

O efeito de filtros passa-baixa é a suavização da imagem, provocando um leve borramento na mesma, pois atenuam ou eliminam os componentes de altas frequências, que correspondem a regiões de bordas (alterações bruscas entre intervalos de níveis de cinza) ou detalhes finos da imagem. Já os filtros passa-alta atenuam ou eliminam os componentes de baixa frequência, realçando as bordas e regiões de alto contraste na imagem (MARQUES FILHO, 1999).

2.2.4.1 Principais técnicas de realce

As principais técnicas de realce utilizadas em domínio espacial e que podem ser usadas também no domínio de frequência, segundo Foresti (2006), são:

- Equalização
- Processamento ponto a ponto
- Processamento por máscara

Equalização é o processo de equalizar um histograma, ou seja, redistribuir os valores de tons de cinza para obter-se maior uniformidade na distribuição de pixels no gráfico. Para isto dispõe-se de uma função auxiliar, chamada de função de transformação.

A forma mais usual de equalizar um histograma é utilizar a função de distribuição acumulada (*cdf – cummulative distribution function*) da distribuição de probabilidades original, que pode ser expressa pela equação 5 (MARQUES FILHO, 1999, p. 61):

$$s_k = T(r_k) = \sum_{j=0}^{k} \frac{nj}{n} = \sum_{j=0}^{k} p_r(r_j) \quad (5)$$

onde:

- $0 \leq r_k \leq 1$ e $k = 0, 1, ..., L-1$
- n_k corresponde ao número de pixels para o nível de cinza correspondente (r_k)
- $p_r(r_k)$ corresponde, no histograma, ao percentual de pixels por nível de cinza

O processamento **ponto a ponto**, segundo Marques Filho (1999), utiliza o conceito de transformação de intensidade, realizando modificações no histograma, onde o valor de nível de cinza de certo pixel após processamento depende apenas

de seu valor original. Nas técnicas de processamento orientadas a vizinhança, o valor resultante depende também dos pixels que são próximos ao elemento da imagem original.

A função de processamento de realce, que depende apenas do pixel em questão, pode ser expressa pela equação (FORESTI, 2006):

$g(x, y) = R(f(x, y))$ onde:

- $g(x, y)$ representa a imagem processada,
- $f(x, y)$ representa a imagem original,
- *R* representa a função de transferência de níveis de cinza (função de mapeamento)

A função R pode ser ou não linear, porém deve respeitar as condições:

- A função *R* deve retornar um único valor, *g*(*x, y*), para cada valor distinto de *f*(*x, y*)
- $0 \leq R(f(x, y)) \leq 1 \; para \; 0 \leq f(x, y) \leq 1$

Assim, a função R mapeará cada pixel da imagem original f(x,y) e aplicará um novo tom de cinza, obtendo desta forma g(x, y).

No processamento **por máscaras**, cada ponto da imagem processada g(x, y) é definido conforme os valores da vizinhança do ponto na imagem original f(x,y). Diferente do que ocorre no processamento ponto a ponto, o valor dos pixels vizinhos interfere no resultado final do processamento. A máscara é uma pequena matriz bidimensional, onde os valores de seus elementos determinam a natureza do processo. Este processamento recebe o nome de filtragem espacial (FORESTI, 2006).

2.2.5 Morfologia Matemática

O estudo da morfologia concentra-se no conjunto de técnicas utilizadas para manipular a estrutura dos objetos. Pode ser empregada em diversas áreas do processamento de imagens, como no realce, filtragem, segmentação, esqueletização, entre outras (GONZALEZ, 2007).

A morfologia digital permite a descrição ou análise da forma de um objeto digital, e sua linguagem é a teoria dos conjuntos. Operações matemáticas podem ser usadas sobre conjuntos de pixels para ressaltar aspectos específicos sobre formas, permitindo que estas sejam reconhecidas ou contadas.

“A base da morfologia consiste em extrair de uma imagem desconhecida a sua geometria através da utilização da transformação de outra imagem completamente definida” (RIBEIRO, Amado, 2003, p. 18).

Este conjunto ou imagem conhecida, usada na transformação é chamado de elemento estruturante. Por exemplo, o conjunto de todos os pixels pretos em uma imagem binária descreve completamente a imagem (uma vez que os demais pontos só podem ser brancos) (MARQUES FILHO, 1999, p.139). Estes conjuntos representam imagens binárias, onde cada elemento do conjunto é um vetor bidimensional, com as coordenadas *(x, y)* dos pixels pretos (por convenção) que representam esta imagem.

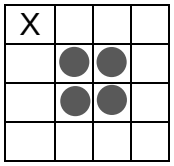
As operações morfológicas básicas são a erosão ($\ominus$) e a dilatação ($\oplus$). A partir da erosão são removidos da imagem os pixels que não atendem a um dado padrão. E a partir da dilatação uma pequena área relacionada a um pixel é alterada para um dado padrão. Porém, dependendo do tipo de imagem a ser processada (em preto e branco, em tons de cinza ou em cores) a definição destas operações é alterada, sendo necessário considerá-las separadamente (DE QUEIROZ; GOMES, 2006).

2.2.5.1 Erosão Binária

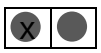
A operação morfológica de erosão binária encolhe a imagem, sua transformação combina dois conjuntos usando vetores de subtração. Ela é expressa como a intersecção de A por B. Tem-se: $A \ominus B = B \cap A$ (RIBEIRO, Amado, 2003). A erosão da imagem A pelo elemento estruturante B pode ser definida como:

$$A \ominus B = (x \mid x + b \in A \text{ para todo } b \in B)$$

Por exemplo, $A \ominus B$:

A 

= {(1,1), (1,2), (2,1), (2,2)}
O “x” representa a origem (0,0) da imagem.

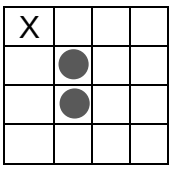
B 

= {(0,0),(1,0)}

O conjunto A $\ominus$ B é o conjunto de translações de B que alinham B sobre o conjunto de pixels pretos em A (RIBEIRO, Amado 2003, p. 21). Não é necessário considerar todas as translações possíveis em B, apenas as que inicialmente possuem a origem de B em um membro de A. São vistas quatro destas para este exemplo:

- $B_{(1,1)}$ = {(1,1), (2,1)} considerando que estes pixels em A são *"pretos"*, o pixel correspondente (1,1) na erosão também o será.
- $B_{(1,2)}$ = {(1,2), (2,2)} considerando que estes pixels em A são *"pretos"*, o pixel correspondente (1,2) na erosão também o será.
- $B_{(2,1)}$ = {(2,1), (3,1)} considerando que o pixel (3,1) em A não é *"preto"*, o pixel (2,1) na erosão não será preto.
- $B_{(2,2)}$ = {(2,2), (3,2)} considerando que o pixel (3,2) em A não é *"preto"*, o pixel (2,2) na erosão não será preto.

Portanto $A \ominus B = \{(1,1) \mid B_{(1,1)} \subseteq A\} \cup \{(1,2) \mid B_{(1,2)} \subseteq A\}$

$A \ominus B =$ 

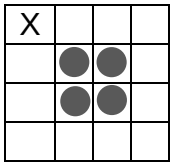
2.2.5.2 Dilatação Binária

A operação morfológica de dilatação, ou *dilação*, combina dois conjuntos usando a adição vetorial. Seu resultado *"engorda"* a imagem, sendo expressa pela união de A por B. A dilatação de um conjunto A por um conjunto B é definida por (RIBEIRO, Amado 2003):

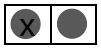
$$A \oplus B = (x \mid x = a + b, a \in A, b \in B)$$

Onde, A representa a imagem sofrendo a operação de dilatação e B representa um segundo conjunto, o elemento estruturante, que definirá a natureza da expansão da imagem A.

Por exemplo, A ⊕ B:

A  = = {(1,1), (1,2), (2,1), (2,2)}

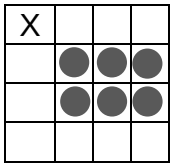
O “x” representa a origem (0,0) da imagem.

B  = = {(0,0),(1,0)}

O conjunto A ⊕ B = {A + (0,0)} ∪ {A + (1,0)} ou seja:

A ⊕ B = {(1,1), (1,2), (2,1), (2,2), (2,1), (2,2), (3,1), (3,2)} organizando sem repetições:

A ⊕ B = {(1,1), (1,2), (2,1), (2,2), (3,1), (3,2)}

A ⊕ B = 

2.2.6 Segmentação

Existem diversos métodos ou técnicas para segmentar imagens, como crescimento de regiões, limiarização, divisão, fusão, agrupamento (ou *clustering*). A escolha do melhor método a ser utilizado depende do problema. Os algoritmos para segmentação de imagens monocromáticas seguem, geralmente, duas estratégias, baseadas na descontinuidade ou na similaridade dos valores de níveis de cinza (TAKAHASHI; BEDREGAL; LYRA, 2005).

Em relação a ideia de descontinuidade, o algoritmo busca dividir a imagem em regiões, de acordo com mudanças bruscas do nível de intensidade luminosa em seus pixels, que podem ser percebidos, por exemplo, em cantos e arestas de objetos na imagem (GONZALEZ, 2007; MERTES, 2012).

Com relação a ideia de similaridade, o algoritmo busca dividir a imagem procurando por regiões que possuam algum padrão de similaridade, como por exemplo, o mesmo nível de intensidade luminosa, a mesma cor ou a mesma textura (GONZALEZ, 2007; MERTES, 2012).

Geralmente um primeiro passo na análise de imagens é a segmentação. Na análise são abordadas técnicas para a extração de informações da imagem, sendo a segmentação a etapa onde técnicas subdividem a imagem em suas partes ou objetos constituintes. O nível até o qual a subdivisão deverá ser realizada depende do problema a ser resolvido. Normalmente automatizar esta fase é uma das tarefas mais difíceis do processamento de imagens (GONZALEZ, 2007).

A abordagem dos algoritmos de segmentação que consideram a descontinuidade é particionar a imagem baseando-se em mudanças bruscas nos níveis de cinza. Suas principais áreas de interesse são a detecção de pontos isolados e detecção de linhas e bordas na imagem. Já a abordagem para os algoritmos que consideram a similaridade baseia-se em limiarização, crescimento de regiões e divisão e fusão de regiões.

2.2.6.1 Detecção de Descontinuidades

As formas básicas de descontinuidades para detecção são pontos, linhas e bordas, que usualmente são localizadas por processos de varredura de imagem por máscaras.

A detecção de pontos isolados em uma imagem pode ser obtida de maneira direta. A Figura 9 representa uma máscara para esta detecção.

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 9 – Máscara para detecção de pontos isolados
Fonte: Gonzalez, 2007.

A máscara percorre a imagem e é considerado encontrado um ponto isolado se |P| > T, onde T representa um limiar não negativo e P é dado pela equação 6:

$$P = W_1 . Z_1 + W_2 . Z_2 + \cdots + W_9 . Z_9 = \sum_{i=1}^{9} W_i . Z_i \quad (6)$$

em que Z_i representa o nível de cinza do pixel da imagem original associado ao coeficiente W_i da máscara. Sendo considerada a vizinhança de conectividade B8.

Então, a Resposta P da máscara é dada, em qualquer ponto da imagem, pela soma de produtos: $\sum_{i=1}^{9} W_i . Z_i$

W1	W2	W3
W4	W5	W6
W7	W8	W9

Figura 10 – Representa uma máscara genérica 3x3
Fonte: Gonzalez, 2010.

A cada posição relativa da máscara sobre a imagem, o pixel central da sub-imagem em questão será substituído, em uma matriz denominada imagem destino, pela soma dos produtos dos coeficientes com os níveis de cinza contidos na sub-região envolvida pela máscara.

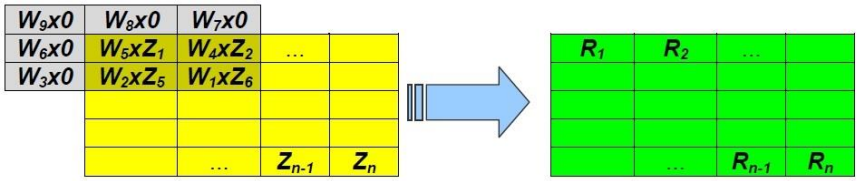


Figura 11 – Processo de convolução com máscara e resultado
Fonte: Neves, 2001.

A detecção de linhas é um pouco mais complexa, pois é necessário localizar os pixels semelhantes e verificar se compõem uma linha comum. Neste caso, são utilizadas máscaras específicas que realçam a existência de linhas na imagem original.

-1	-1	-1	-1	-1	2	-1	2	-1	2	-1	-1
2	2	2	-1	2	-1	-1	2	-1	-1	2	-1
-1	-1	-1	2	-1	-1	-1	2	-1	-1	-1	2

Figura 12 – Máscara horizontal, de 45º, vertical e de135º, respectivamente.
Fonte: Gonzalez, 2010.

Considerando que P_1, P_2, P_3 e P_4 são as repostas das máscaras da Figura 12, sendo fornecidos a partir da fórmula $\sum_{i=1}^{9} W_i . Z_i$ destas máscaras aplicadas a uma imagem. Se um certo ponto $|P_i| > |P_j|$ para todos os $j \neq i$, então diz-se que este ponto está mais associado a uma linha na direção da máscara *i*. Por exemplo, se em um determinado ponto da imagem tem-se $|P_1| > |P_j|$, para $j = 2, 3,4$, então aquele ponto em particular está mais associado a uma linha horizontal (GONZALEZ, 2010).

A detecção de bordas é a abordagem mais comum para a detecção de descontinuidades significantes nos níveis de cinza, provavelmente devido a baixa ocorrência de pontos e linhas isoladas em aplicações práticas (GONZALEZ, 2010). “Uma borda é um limite entre duas regiões com propriedades relativamente distintas de nível de cinza” (GONZALEZ, 2010, p. 297).

Para a **detecção de bordas** são aplicáveis os filtros espaciais lineares baseados no gradiente da função de luminosidade da imagem e os baseados no laplaciano.

Pode-se descrever a detecção de bordas com a utilização de derivadas. Uma interpretação de derivada seria considerá-la como a taxa de mudança de uma

função, e a taxa de mudança dos níveis de cinza em uma imagem é maior perto das bordas e menor em áreas constantes. Assim, se forem tomados os valores da intensidade da imagem e encontrarem-se os pontos onde a derivada é um ponto de máximo, assim tem-se identificadas as suas bordas.

Considerando que a imagem digital é uma função bidimensional f(x,y), é importante considerar mudanças nos níveis de cinza em muitas direções. Por isso são utilizadas as derivadas parciais da imagem nas direções *x* e *y*. Uma estimativa da direção atual da borda pode ser obtida com o operador gradiente, muito utilizado no processamento de imagens, expresso por (NASCIMENTO, 2013):

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\delta f}{\delta x} \\ \frac{\delta f}{\delta y} \end{bmatrix}$$

A magnitude do vetor gradiente, denotada por $mag\,(\nabla f)$, corresponde ao valor da taxa de variação na direção do vetor gradiente, na equação 7:

$$mag(\nabla f) = \sqrt{G_x^2 + G_y^2} = \sqrt{\left(\frac{\delta f}{\delta x}\right)^2 + \left(\frac{\delta f}{\delta y}\right)^2} \quad (7)$$

O ângulo de direção do vetor ∇f é dado por: $\alpha(x,y) = tan^{-1}\left(\frac{G_y}{G_x}\right)$, onde o ângulo α é medido em relação ao eixo x. A direção da borda em um ponto(x,y) é sempre perpendicular à direção $\alpha(x,y)$ do vetor gradiente, conforme ilustrado na Figura 13.

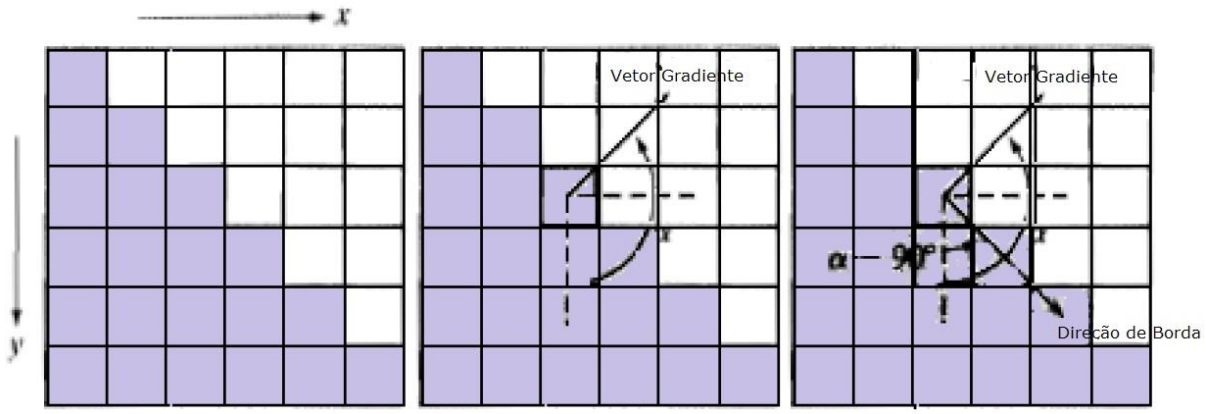


Figura 13 – Determinação de intensidade e direção de borda com vetor gradiente
Fonte: Nascimento, 2013.

Como uma imagem digital é uma matriz formada por um conjunto discreto de pixels, ela não pode ser diferenciada da maneira usual, pois utiliza derivadas parciais para o cálculo do gradiente. As derivadas são aproximadas por diferenças

entre pixels de uma determinada região e podem ser calculadas de formas diferentes.

Uma aproximação muito utilizada para as derivadas parciais de primeira ordem discreta é dada por:

$$\frac{\delta f}{\delta x}(x,y) \approx f(x+1,y) - f(x,y)$$

$$\frac{\delta f}{\delta y}(x,y) \approx f(x,y+1) - f(x,y)$$

Estas aproximações podem ser implementadas com filtros utilizando a máscara 1-D, como as ilustradas na Figura 14.

-1	1

-1
1

Figura 14 – Máscaras unidimensionais
Fonte: Nascimento, 2013.

Um algoritmo de detecção de bordas, conhecido por operador **Roberts**, utiliza uma máscara 2x2 para localizar as mudanças de direção em x e y. O gradiente para determinar se o pixel avaliado é um pixel de borda é calculado por $|G| = \sqrt{G_x^2 + G_y^2}$ (VON WANGENHEIM, 2008).

0	1
-1	0

1	0
0	-1

Figura 15 – Máscaras de Roberts para G_x e G_y
Fonte: Mertes, 2012

Se a magnitude calculada for maior que o menor valor de entrada pré-definido, o pixel é considerado como parte da borda. A direção do gradiente da borda, perpendicular a direção da borda, é definida por: $\alpha(x,y) = tan^{-1}\left(\frac{G_y}{G_x}\right)$, como exposto anteriormente.

O pequeno tamanho da máscara de Roberts torna este operador de fácil implementação e rápido no cálculo da máscara de resposta. Porém estas respostas são consideradas muito sensíveis aos ruídos presentes na imagem.

Outro algoritmo para detecção de bordas, o operador de **Sobel,** é frequentemente usado para detecção de bordas horizontais e verticais, através do cálculo dos gradientes da função em relação ao elemento central da máscara. Sua implementação computacional é simples e pode ser especificada a partir de máscara 3x3 conforme a Figura 16 (MERTES, 2012).

-1	0	1
-2	0	2
-1	0	1

1	2	1
0	0	0
-1	-2	-1

Figura 16 – Máscaras de Sobel para G_x e G_y
Fonte: Mertes, 2012.

O operador de Sobel utiliza as mesmas fórmulas que o operador de Roberts para encontrar o gradiente e o ângulo. Esta técnica é menos suscetível ao ruído, por utilizar uma máscara de dimensão maior. Seus resultados são considerados mais precisos. Porém a computação do gradiente (∇f) torna-se mais complexa. Na prática o gradiente é aproximado da seguinte forma: $\nabla f = \nabla x + \nabla y$.

Um algoritmo para detecção de bordas baseado em funções de **Laplace**, é definido por uma derivada de segunda ordem dada por: $\nabla^2 f(x,y) = \frac{\delta^2}{\delta x^2} + \frac{\delta^2}{\delta y^2}$.

Da mesma forma como ocorre em relação aos gradientes, a derivada de segunda ordem precisa ser aproximada para uma forma discreta, e para isso podem ser usadas as equações:

$$\frac{\delta^2 f}{\delta x^2}[x,y] \approx f[x+1,y] + f[x-1,y] - 2f[x,y]$$

$$\frac{\delta^2 f}{\delta y^2}[x,y] \approx f[x,y+1] + f[x,y-1] - 2f[x,y]$$

Estas equações podem ser combinadas, tendo-se a equação 8:

$$\nabla^2 f(x,y) = f[x+1,y] + f[x-1,y] + f[x,y+1] + f[x,y-1] - 4f[x,y] \quad (8)$$

A máscara bidimensional para implementar a aproximação do operador Laplaciano pode ser visualizada na Figura 17.

0	-1	0
-1	4	-1
0	-1	0

Figura 17–Máscara Laplaciana
Fonte: Nascimento, 2013.

Embora o operador Laplaciano seja insensível à rotação e, portanto, capaz de realçar ou detectar bordas em qualquer direção, ele é pouco utilizado para detecção de bordas, pois ao utilizar derivadas de segunda ordem, ele tende a detectar com muita facilidade mudanças bruscas, o tornando sensível ao ruído de forma inaceitável (NASCIMENTO, 2013; NEVES, 2001).

Neste capítulo foram vistos conceitos de imagem, de distâncias entre pixels, operações sobre imagens com máscaras que serão relevantes no entendimento

deste trabalho ao gerarmos imagens intervalares ou sobre o processo de convolução.

2.3 IMAGENS DIGITAIS INTERVALARES

No processamento de imagens digitais podem surgir erros ocasionados durante a aquisição ou no processo de discretização da imagem original (contínua) para a imagem digital (discreta). Estes erros usualmente levam a incertezas com relação à intensidade de brilho dos pixels da imagem, causando importantes perdas de informações para aplicações que necessitem de precisão na análise da imagem (CRUZ, 2008).

Diversos fatores podem influenciar na aquisição de uma imagem ou em seu processamento, trazendo divergências na interpretação (quantização) dos elementos da imagem (pixels). Por exemplo, a aquisição de uma mesma imagem por dispositivos eletrônicos diferentes pode ocasionar inexatidão nas respectivas imagens digitais obtidas. Além disso, existem definições imprecisas na análise, como conceitos de extremidade, cantos ou relações entre regiões que podem aumentar as incertezas no processo (LYRA et al., 2003).

A área de processamento de imagens tem evoluído continuamente e tem sido aprimorada para possuir maior controle sobre a existência de erros no processo de análise da imagem, podendo utilizar metodologias diferentes neste intento. A matemática intervalar se apresenta como uma poderosa ferramenta para análise e controle de erros computacionais, considerando os números pontuais como intervalos que os encapsulam, armazenando dados físicos por uma medida provável, com porcentual de erro associado (CRUZ et al., 2010).

Cada pixel, em uma imagem digital, pode representar um intervalo matemático natural. Isto ocorre em decorrência da variação das condições de captura e tratamento da imagem, como: iluminação ambiental, reflectância dos objetos, regulagem de aparelhos, software utilizado no tratamento, etc (LYRA et al., 2003).

Para minimizar a perda de dados, Lyra et al. (2003) introduziram em “Imagens Digitais Intervalares” uma abordagem, baseada na matemática intervalar, de controle de informações e possíveis erros no momento de captura e processamento, desenvolvendo técnicas e conceitos matemáticos que deram suporte a sua teoria (RIBEIRO; LYRA, 2004).

A matemática intervalar, proposta por Moore, na década de 60, tem sido muito utilizada associada a diversas áreas, buscando um controle rigoroso em

diversos tipos de erros computacionais, que envolvam representações finitas de números reais, como ocorre na discretização de imagens (TAKAHASHI; BEDREGAL; LYRA, 2005).

Uma imagem digital intervalar possui valores intervalares em seus pixels, ao invés de valores pontuais. Considerando não haver dispositivo físico que faça a captura destas imagens intervalares diretamente, Lyra et al. (2003) desenvolveram um aplicativo para obtenção e processamento de imagens intervalares com técnicas estendidas para o processamento intervalar (LYRA et al., 2003; LYRA et al., 2004; TAKAHASHI; BEDREGAL; LYRA, 2005).

Em seu trabalho, Lyra et al. (2003) utilizaram, para obtenção das imagens, as técnicas: captura por dispositivos diferentes; captura com parâmetros de iluminação e reflectância de objetos diferentes; e captura em movimento. Então, foi criada a imagem intervalar baseada em sua vizinhança de 8. Para a geração destas imagens, neste aplicativo, pôde-se optar pelos valores ínfimos e supremos de cada elemento RGB ou os valores ínfimos e supremos da luminância (obtida pela média inteira dos 3 valores RGB) (LYRA et al., 2003; RIBEIRO; LYRA, 2004).

Ribeiro e Lyra (2004) diferenciam melhor as duas técnicas utilizadas em sua abordagem para a criação da imagem intervalar. Em uma das técnicas o pixel intervalar é gerado a partir de valores de seus pixels vizinhos, ou seja, cada pixel da imagem é um intervalo formado pela maior e menor intensidade de seus vizinhos. Na outra técnica adotada, a imagem é baseada em várias imagens do mesmo objeto, obtidas através de dispositivos de captura diferentes, ou com configurações diferentes do mesmo dispositivo, ou ainda com condições ambientais diferentes (RIBEIRO; LYRA, 2004).

Lyra et al. (2004), reforçaram o conceito de imagens, definindo-a como sendo uma função luminosa bidimensional, denotada por $f(x,y)$, em que o valor ou a amplitude de f nas coordenadas espaciais (x, y) dá a intensidade (brilho) da imagem no mesmo ponto. Situaram que para o processamento digital, a imagem deve ser digitalizada, tanto espacialmente quanto em amplitude. E que isto ocorre nos processos de amostragem e quantização (níveis de cinza), que formam o estágio onde ocorre a aproximação de uma imagem contínua em amostras igualmente espaçadas em forma de matriz, onde cada elemento é denominado por pixel.

Esta discretização ocorre duas vezes, conforme Lyra et al. (2004) explicam, para criar o espaço bi-dimensional da imagem, em forma de matriz e novamente

para informar os valores de intensidade de tons de cinza em cada posição de pixel. Para abordarem a introdução da versão intervalar nesta fase, trazem a informação dos erros que podem surgir nesta etapa, por melhor que seja a resolução utilizada na abordagem, pois uma imagem digital (discreta) nunca corresponderá a uma imagem existente na natureza.

Lyra et al. (2004) afirmam que a matemática intervalar por sua vez, traria uma melhoria certa com sua representação de pixels de forma contínua (um intervalo é um espaço contínuo), com a certeza do valor atribuído ao pixel estar contido no intervalo tão grande quanto desejado, muitas vezes não viável de ser computado.

Na abordagem intervalar, conforme Cruz e Santiago (2011), pode ser atribuída na forma de erros, $\varepsilon > 0$, com relação aos valores pontuais de entrada dos pixels (intervalização), ou com auxílio de um conjunto de imagens, considerando o menor e o maior valor correspondente a cada pixel (característica peculiar de modelos intervalares). Após esta etapa de intervalização, todo o processamento da imagem intervalar possui um controle dos erros aplicando-se operações intervalares corretas. A saída esperada não é pontual, e sim intervalar, onde cada pixel codifica o erro cometido durante todo o processamento. Assim, o controle dos erros (incertezas) cometidos durante o processamento intervalar é mais eficiente que na abordagem fuzzy, uma vez que não tem a introdução de métodos de defuzzificação (CRUZ; SANTIAGO, 2011).

Segundo Kearfott e Kreinovich (1996), o poder da aritmética intervalar vem de dois fatos: por suas operações elementares e funções padrão poderem ser computadas para intervalos usando fórmulas e subrotinas simples e pelos arredondamentos direcionados que podem ser usados, de forma que a imagem destas operações contenha rigorosamente o exato resultado da computação. Estes fatos permitem um encapsulamento rigoroso do erro de arredondamento, do erro de truncamento e erros nos dados, e também o cálculo rigoroso de limites dos intervalos das funções (KEARFOTT; KREINIVICH, 1996).

2.3.1 Morfologia Matemática

A Morfologia Matemática é uma das áreas de destaque do processamento de imagens com inúmeras práticas de análise de imagens. Seus principais operadores são a dilatação e a erosão, sob a ótica intervalar, que se apresentam como boas

ferramentas para lidar com questões de incertezas nas imagens (CRUZ; SANTIAGO, 2011).

2.3.1.1 Conjunto de intervalos naturais

Um conjunto de intervalos naturais pode ser definido por:

Seja N o conjunto dos naturais. O conjunto $\mathbb{IN} = \{[a,b]\}: a,b \in \mathbb{N}: a \leq b\}$.

Seja $n \in$ N e $N = [n,n]$. $\Omega_{\mathbb{N}} = \{[x,y] \in \mathbb{IN}: x \leq y \leq n\}$ é o conjunto de pixels intervalares.

Um intervalo $X \in \Omega_{\mathbb{N}}$ pode ser denotado por $X = [\underline{x}, \overline{x}]$ ou, por simplicidade $X = [x_1, x_2]$.

Ω_N é um reticulado completo, ou seja, é um conjunto parcialmente ordenado no qual existe ínfimo e supremo de cada subconjunto não vazio de elementos (CRUZ; SANTIAGO, 2011).

2.3.1.2 Reticulado completo de imagens intervalares sobre $\Omega_{\mathbb{N}}$

As funções $F: E \to \Omega_{\mathbb{N}}$ modelam imagens intervalares em tons de cinza, considerando que $\Omega_{\mathbb{N}}$ é um reticulado completo e E é um conjunto de coordenadas de números inteiros, $E = \mathbb{Z}$ x $\mathbb{Z}$.

Uma imagem intervalar em tons de cinza, cuja posição dos pixels correspondem a valores intervalares, é definida pelo mapeamento $F: E \to \Omega_{\mathbb{N}}$. O conjunto de todas as funções que representam uma imagem intervalar em escalas de cinza será denotado por $\Omega_{\mathbb{N}}^{\mathrm{E}}$.

Tendo-se F, G $\in \Omega_{\mathbb{N}}^{\mathrm{E}}$, a relação de ordem a seguir pode ser definida sobre $\Omega_{\mathbb{N}}^{\mathrm{E}}$:

$$F \leq G \iff F(x) \leq G(x), \forall\, x \in E$$

Ou seja, $\langle \Omega_{\mathbb{N}}^{\mathrm{E}}, \leq \rangle$ torna-se uma ordem parcial.

Desde que Ω_N é um retículo completo, $\Omega_{\mathbb{N}}^{\mathrm{E}}$ também é um reticulado com ordem parcial $\leq$ e operações de supremo e ínfimo (CRUZ et al., 2010; CRUZ; SANTIAGO, 2011).

2.3.1.3 Operações de supremo e ínfimo

Dadas duas funções intervalares F, G $\in \Omega_{\mathbb{N}}^{\mathrm{E}}$. As operações de supremo e ínfimo entre F e G são definidas, respectivamente, por (CRUZ et al., 2010; CRUZ; SANTIAGO, 2011):

$(F \vee G)_{(x)} = F_{(x)} \vee G_{(x)}$

$(F \wedge G)_{(x)} = F_{(x)} \wedge G_{(x)}$

Assim, se $F_{(x)}$= [a; b] e $G_{(x)}$ = [c; d] então

$(F \vee G)_{(x)}$ = [max (a,c); max(b,d)] *para intervalo supremo de F e G*

$(F \wedge G)_{(x)}$ = [min (a,c); min(b,d)] *para intervalo ínfimo de F e G*

Por exemplo: $F_{(x)}$ = [3; 10]e $G_{(x)}$ = [4; 9]

$(F \vee G)_{(x)}$ = [max (3,4); max(10,9)] = [4, 10]

$(F \wedge G)_{(x)}$ = [min (3,4); min(10,9)] = [3, 9]

2.3.1.4 Dilatação e Erosão para imagens intervalares em níveis de cinza

Estes operadores utilizam conceitos de translação, reflexão, soma e diferença de Minkowiski, que serão abordados a seguir. A translação pode ser vista como um deslocamento paralelo da imagem selecionada, em função de um vetor, preservando direção, comprimento de segmento de reta e a amplitude dos ângulos (CRUZ et al., 2010; CRUZ; SANTIAGO, 2011).

1. Translação horizontal:

Dada uma imagem intervalar $F: E \to \Omega_{\mathbb{N}}$, $u \in E$. A translação horizontal de F por u, é a função intervalar $F_u: E \to \Omega_{\mathbb{N}}$ definida por $F_u(x) = F(x - u)$.

2. Translação vertical:

Dada uma imagem intervalar $F: E \to \Omega_{\mathbb{N}}$, $V \in E$. A translação vertical de F_x por V, é definida por $(F + V)_{(x)} = F_{(x)} + V$.

3. Translação morfológica intervalar

Quando ambas as translações são aplicadas juntas tem-se uma translação morfológica intervalar, dada por: $(F_u + V)_{(x)} = F_{(x-u)} + V$

A Figura 18 apresenta um gráfico de uma função intervalar com uma translação morfológica.

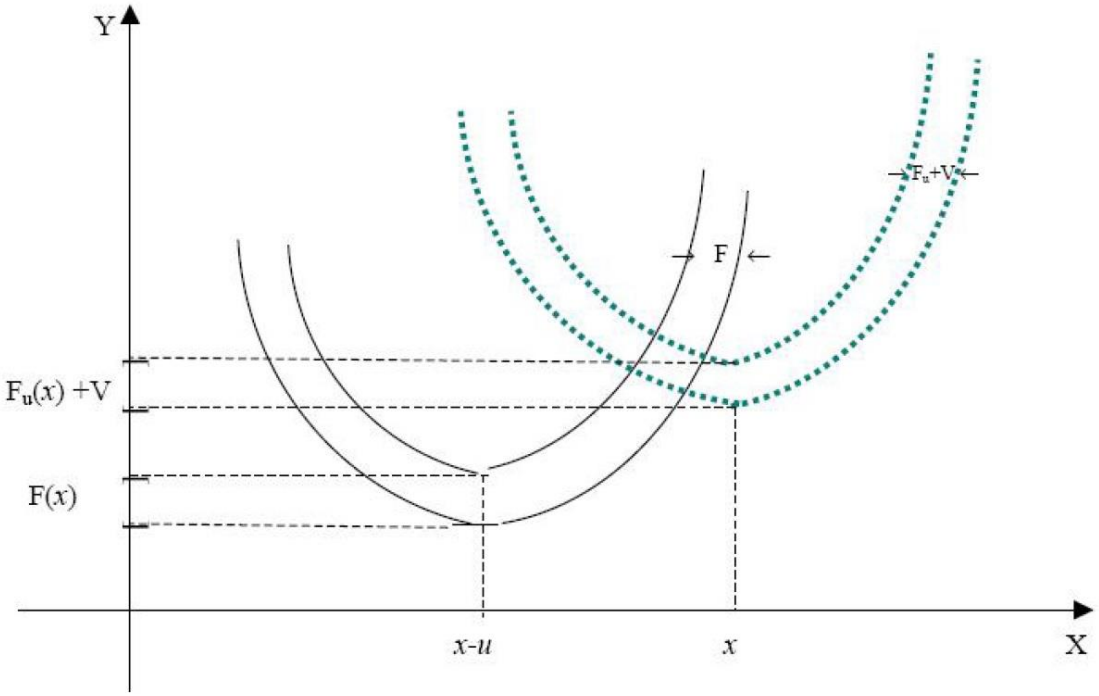


Figura 18 – Função intervalar com translação morfológica
Fonte: Cruz, 2008, página 98.

4. Reflexão

Dada uma imagem intervalar G $\in \Omega_{\mathbb{N}}^{E}$ $x \in E$, a reflexão de G é definida por:

$$\hat{G}(x) = G(-x).$$

Observando-se que: $F_{-u}(x) = F(x+u)$

Seja $F: E \to \Omega_{\mathbb{N}}$ então existe $\underline{F}, \overline{F}$: E → L tal que, $F(u) = [\underline{F}(u); \overline{F(u)}]$.

Ou seja, a função intervalar F pode ter uma forma intervalar, onde as extremidades são funções que pertencem ao caso pontual (CRUZ et al., 2010; CRUZ; SANTIAGO, 2011).

5. ***Soma e Diferença de Minkowiski***

Considerando duas imagens intervalares $F, G \in \Omega_{\mathbb{N}}^{E}$. A soma e a diferença Minkowiski são definidas, respectivamente, por:

$$F \ominus G = \bigwedge_{u \,\epsilon\, dom(G)} \left(Fu - \hat{G}(u) \right)$$

$$F \oplus G = \bigvee_{u \,\epsilon\, dom(G)} \left(Fu + G(u) \right)$$

6. ***Dilatação e Erosão***

Em geral as operações de dilatação e de erosão, para imagens em tons de cinza, são definidas pelas equações 9 e 10 (CRUZ et al., 2010; CRUZ; SANTIAGO, 2011):

Sendo duas funções F e G, então:

$$\Delta_G(F)_{(x)} = (F \oplus G)_{(x)} = \bigvee_{u\,\epsilon\,dom(G)} [F(x-u) + G(u)] \quad (9)$$

$$\varepsilon_G(F)_{(x)} = (F \ominus G)_{(x)} = \bigwedge_{u\,\epsilon\,dom(G)} [F(x-u) - \hat{G}(u)] \quad (10)$$

A dilatação e erosão correspondem a $\Delta_G(F)$ e $\varepsilon_G(F)$, respectivamente. G é chamada de função aditiva estruturante.

Considerando que $\hat{G}(x) = G(-x)$, para o caso de erosão, pode-se escrever sua equação também pela equação 11:

$$\varepsilon_G(F)_{(x)} = (F \ominus G)_{(x)} = \bigwedge_{u\,\epsilon\,dom(G)} [F(x+u) - G(u)] \quad (11)$$

Em Cruz (2008) ainda são apresentadas outras formas alternativas para a definição da dilatação e da erosão, com detalhamentos específicos da morfologia, podendo ser de interesse para aprofundamento em estudos mais teóricos destas operações.

Um exemplo foi fornecido por Cruz (2008) de uma imagem intervalar descrita em uma grade E = 8x8, com operações de dilatação e erosão agindo sobre esta imagem, por uma função estruturante G pré-definida. Em Cruz et al. (2010) também é fornecido um exemplo destas operações, porém com grade E = 10x10.

Considerou-se que $F \in \Omega_{\mathbb{N}}^{\mathrm{E}}$, tal que F=[$f_1$; f_2], onde f_1 seria o mínimo do intervalo e f_2 o máximo. Supôs que uma sequência de imagens foi obtida pela mesma posição e representadas pelo conjunto de pares ordenados dados por {(0,0), (1,5), (2,3), (3,4), (4,2), (5,5), (6,6)}. Para cada posição, havia um valor que algumas vezes apresentava incerteza. De acordo com a definição de imagem intervalar, tomou-se para cada intervalo de F, o menor e o maior elemento. Ficando seus valores, definidos arbitrariamente:

$f_1(0,0) = 1$ e $f_2(0,0) = 3$,

$f_1(1,5) = 4$ e $f_2(1,5) = 6$,

$f_1(2,3) = 2$ e $f_2(2,3) = 4$,

$f_1(3,4) = 0$ e $f_2(3,4) = 2$,

$f_1(4,2) = 5$ e $f_2(4,2) = 5$,

$f_1(5,5) = 6$ e $f_2(5,5) = 7$,

$f_1(6,6) = 6$ e $f_2(6,6) = 8$.

Formando assim o conjunto:

Im(F) = {[1,3]; [4,6]; [2,4]; [0,2]; [5,5]; [6,7]; [6,8]} ou seja, é formada uma função intervalar $F: E \rightarrow \Omega_8$ ou em ordem:

Im(F) = {[0,2]; [1,3]; [2,4]; [4,6]; [5,5]; [6,7]; [6,8]}

Foi considerado G como uma função estruturante e pontual, ou seja,$g_1 = g_2$, para G = [g_1, g_2].

Sendo que $u \epsilon \mathbb{Z}$ tal que, $u = \{u_1, u_2, u_3\}$, onde $u_1 = (0,0), u_2 = (0,2)$e $u_3 = (1,2)$, no exemplo citado.

A Figura 19 representa a imagem intervalar F e a função estruturante G para $G(u_1) = [2,2], G(u_2) = [1,1]$ e $G(u_3) = [3,3]$.

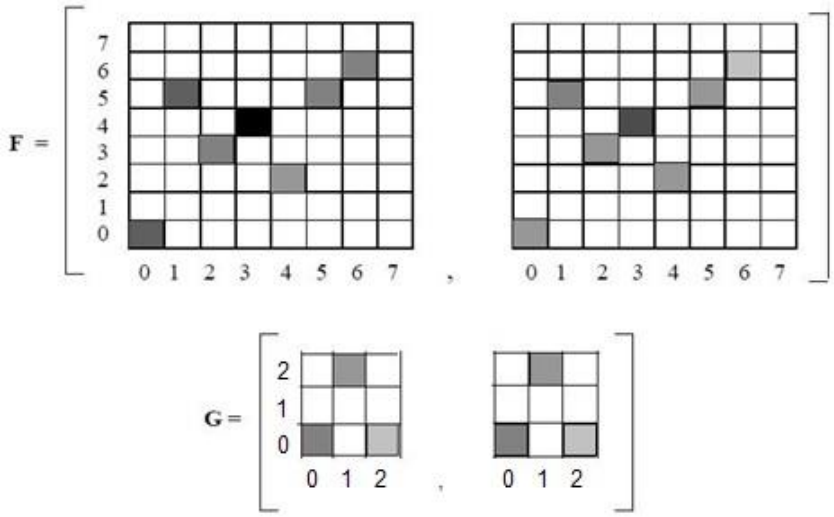


Figura 19 – Imagem representada por função *F* e por função estruturante *G*
Fonte: Cruz, 2008, página 106.

Cruz (2008) para calcular a dilatação, primeiro calculou todas as translações horizontais da função F, para cada u_i, *i* = 1,2,3. Em seguida, calculou $F(x-u) + G(u), x \in F$ e finalmente aplicou o *supremo*, ou seja, calculou $\bigvee [F(x-u) + G(u)]$. Obteve aparentemente os mesmos valores nas translações de F por u_1, F por u_2 e F por u_3, porém em posições diferentes ou melhor, adicionou posições à nova imagem.

Translação de F por u_1 = (0,0)

$F_{u1}(0,0)$ = F((0,0) - (0,0)) = F(0,0) = [1,3]

$F_{u1}(1,5)$ = F((1,5) - (0,0)) = F(1,5) = [4,6]

$F_{u1}(2,3)$ = F((2,3) - (0,0)) = F(2,3) = [2,4]

$F_{u1}(3,4)$ = F((3,4) - (0,0)) = F(3,4) = [0,2]

$F_{u1}(4,2)$ = F((4,2) - (0,0)) = F(4,2) = [5,5]

$F_{u1}(5,5)$ = F((5,5) - (0,0)) = F(5,5) = [6,7]

$F_{u1}(6,6)$ = F((6,6) - (0,0)) = F(6,6) = [6,8]

Translação de F por u_2 = (0,2)

$F_{u2}(0,2)$ = F((0,2) - (0,2)) = F(0,0) = [1,3]

$F_{u2}(1,7)$ = F((1,7) - (0,2)) = F(1,5) = [4,6]

$F_{u2}(2,5)$ = F((2,5) - (0,2)) = F(2,3) = [2,4]

$F_{u2}(3,6)$ = F((3,6) - (0,2)) = F(3,4) = [0,2]

$F_{u2}(4,4)$ = F((4,4) - (0,2)) = F(4,2) = [5,5]

$F_{u2}(5,7)$ = F((5,7) - (0,2)) = F(5,5) = [6,7]

$F_{u2}(6,8)$ = F((6,8) - (0,2)) = F(6,6) = [6,8]

Translação de F por u_3 = (1,2)

$F_{u3}(1,2)$ = F((1,2) - (1,2)) = F(0,0) = [1,3]

$F_{u3}(2,7)$ = F((2,7) - (1,2)) = F(1,5) = [4,6]

$F_{u3}(3,5)$ = F((3,5) - (1,2)) = F(2,3) = [2,4]

$F_{u3}(4,6)$ = F((4,6) - (1,2)) = F(3,4) = [0,2]

$F_{u3}(5,4)$ = F((5,4) - (1,2)) = F(4,2) = [5,5]

$F_{u3}(6,7)$ = F((6,7) - (1,2)) = F(5,5) = [6,7]

$F_{u3}(7,8)$ = F((7,8) - (1,2)) = F(6,6) = [6,8]

As novas posições são dadas pelo conjunto: {(0,0), (1,5), (2,3), (3,4), (4,2), (5,5), (6,6), (0,2), (1,7), (2,5), (3,6), (4,4), (5,7), (**6,8**), (1,2), (2,7), (3,5), (4,6), (5,4), (6,7), (**7,8**)}

Obs.: Cruz (2008, p.106) trazia a observação: *"Note que, nas posições (6,8) e (7,8) toma-se (6,7) e (7,7)"* que aparentemente referiam-se aos endereços do novo conjunto, assim presente na bibliografia: {(0,0), (1,5), (2,3), (3,4), (4,2), (5,5), (6,6), (0,2), (1,7), (2,5), (3,6), (4,4), (5,7), (6,7), (1,2), (2,7), (3,5), (4,6), (5,4), (6,7), (7,7)}

Então, Cruz (2008) calculou a soma de Minkowiski, $F(x-u) + G(u)$.

F(0,0) + G(0,0) = [1,3] + [2,2] = [3,5]

F(1,5) + G(0,0) = [4,6] + [2,2] = [6,8]

F(2,3) + G(0,0) = [2,4] + [2,2] = [4,6]

F(3,4) + G(0,0) = [0,2] + [2,2] = [2,4]

F(4,2) + G(0,0) = [5,5] + [2,2] = [7,7]

F(5,5) + G(0,0) = [6,8] + [2,2] = [8,8]

F(6,6) + G(0,0) = [**7,7**] + [2,2] = [8,8]

De forma análoga para G(0,2) e G(1,2) foram calculadas as somas de Minkowiski.

Por fim, Cruz (2008) calculou, para cada posição de x, $\bigvee [F(x-u)+G(u)]$. Assim obteve a dilatação intervalar (F $\oplus$ G) que está representada na Figura 20.

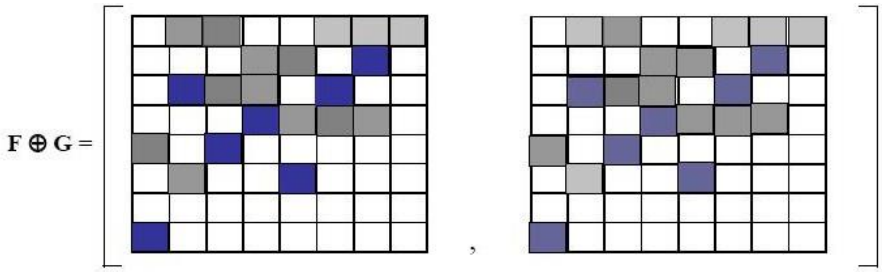


Figura 20 – Dilatação intervalar de F pela função estruturante G
Fonte: Cruz, 2008, página 108.

Cruz (2008) seguiu demonstrando em seu exemplo o cálculo da erosão, $\varepsilon_G(F)_{(x)} = (F \ominus G)_{(x)} = \bigwedge_{u \in E} [F(x+u) - G(u)]$ e a Figura 21 demonstra a erosão intervalar obtida pelas translações da função estruturante G sobre a imagem intervalar F: E $\rightarrow \Omega_8$.

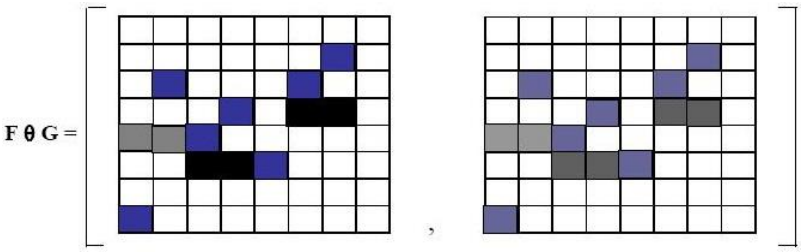


Figura 21 – Erosão intervalar de F pela função estruturante de G
Fonte: Cruz, 2008, página 110.

Pode-se notar que os pixels mais escuros estão representados à direita da imagem intervalar e os mais claros, à esquerda. No exemplo visto, também é possível notar que a imagem intervalar difere pelo tom de cinza. Quando os tons são diferentes à esquerda e à direita da imagem, na mesma posição, significa que, nesta posição existe uma incerteza.

Uma importante propriedade algébrica dos operadores elementares da morfologia matemática é a relação da dualidade entre erosão e dilatação $\neg(F \oplus G) = \neg(F \ominus G)$ e Cruz (2008) mostra que essa propriedade também vale para o caso intervalar em seu trabalho.

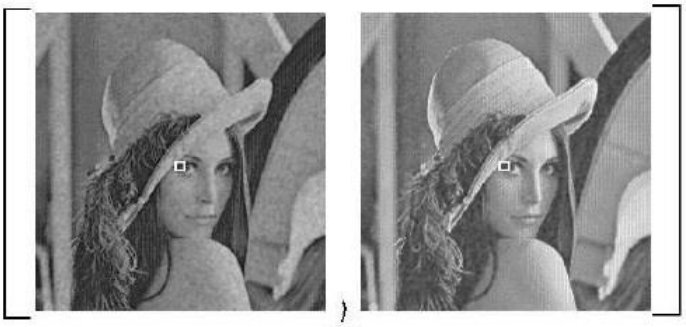
2.3.2 Imagem intervalar representada por Matriz Intervalar

Tem-se que uma imagem digital intervalar, é uma matriz *A* de ordem *m*x*n*, que representa uma imagem espacialmente discretizada e digitalizada em forma intervalar, onde cada tom de cinza torna-se um intervalo cuja diferença entre limite

inferior e superior é um valor aceitável em relação ao lugar no espaço (LYRA et al., 2004).

Cada $\mathbf{a}_{ij}$ em *A* é denominado por pixel intervalar, porque seus valores são intervalos que definem a variação Γ da intensidade de luminosidade deste pixel intervalar (LYRA et al., 2004).

Na Figura 22 o item (a) representa uma imagem digital intervalar e o item (b) representa os pixels intervalares da parte selecionada na imagem intervalar.



(a)

[147,170]	[131,153]	[137,153]	[146,187]	[155,187]	[161,187]
[144,187]	[130,153]	[138,170]	[145,187]	[152,204]	[163,187]
[147,153]	[138,153]	[145,170]	[156,204]	[163,187]	[163,187]
[145,153]	[144,153]	[149,170]	[158,204]	[153,204]	[147,204]
[141,153]	[155,187]	[149,204]	[158,187]	[139,187]	[102,125]
[154,187]	[162,187]	[162,204]	[163,204]	[102,120]	[102,120]

(b)

Figura 22 – Representação de uma imagem digital intervalar
Fonte: Takahashi, Bedregal e Lyra, 2005, página 4.

2.3.2.1 Vizinhanças de pixels intervalares

Seja p um pixel intervalar com coordenadas (x,y).

Os vizinhos (de borda) $\mathbf{B_4}$ de p, denotados por $N_4(p)$ são formados pelos pixels de coordenadas (x+1, y), (x-1, y), (x, y+1), (x, y-1) ou seja, $N_4(p)$ = {(x+1, y), (x-1, y), (x, y+1), (x, y-1)}

Os vizinhos (de diagonal) $\mathbf{B_8}$ de p, denotados por $N_8(p)$ são formados pelos pixels de coordenadas (x+1, y), (x-1, y), (x, y+1), (x, y-1), (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1) ou seja, $N_8(p)$ = {(x+1, y), (x-1, y), (x, y+1), (x, y-1), (x+1, y+1), (x+1, y-1), (x-1, y+1), (x-1, y-1)}

Cada pixel intervalar está em uma unidade de distância de *(x,y)* e alguns vizinhos de p estarão fora da imagem digital se a coordenada *(x,y)*representar uma borda da imagem digital (LYRA et al., 2004).

2.3.2.2 Conectividade entre pixels intervalares

A conectividade é um importante conceito no tratamento de imagens intervalares, que ligam as vizinhanças e níveis de cinza, usada constantemente para estabelecer as bordas dos objetos e os componentes de áreas da imagem (LYRA et al., 2004).

Conectividade entre pixels intervalares p e q. Seja V = [a, b] um intervalo que define a variação de níveis de cinza que satisfaça a certo critério de similaridade:

- Conectividade por 4: p, q $\subseteq$ V, são conectados por 4 se q $\in N_4(p)$.
- Conectividade por 8: p, q $\subseteq$ V, são conectados por 8 se q $\in N_8(p)$.
- Conectividade por m: p, q $\subseteq$ V, são conectados por m se:
  - q $\in N_4(p)$ ou
  - q $\in N_8(p)$ e o conjunto $N_4(p) \cap N_8(q) = \varnothing$

2.3.2.3 Distância em Imagens Intervalares

Sejam $I_p = [p_1, p_2]$ e $I_q = [q_1, q_2]$ as intensidades intervalares dos pixels p e q respectivamente. A distância Moore entre estas intensidades intervalares é dada pela função $D(I_p, I_q) = Max(|p_1-q_1|, |p_2-q_2|)$, que satisfaz a noção clássica de métrica.

Já a distância intervalar desenvolvida por Acioly e Bedregal (1997) é dada pela função $Q(I_p, I_q) = 0$ se $q_1 \leq p_1 \leq p_2 \leq q_2$ e , $Q(I_p, I_q) = Max(q_1 - p_1; p_2 - q_2)$ que satisfaz a noção de quasi-métrica.

Uma quasi-métrica em um conjunto não vazio M é uma função $d: M \cdot M \rightarrow \Re$ que satisfaz (SANTANA, 2012):

1. $d(x,y) \geq 0$ e $d(x,y) = 0$;
2. se $d(x,y) = d(y,x) = 0$, então $x = y$;
3. $d(x,z) \leq d(x,y) + d(y,z)$ (desigualdade triangular)

Se d é uma quasi-métrica então pode ocorrer $d(x,y) = 0$, com $x \neq y$;

As noções de distâncias Euclidiana, de Manhattan (ou quarteirão) e do Tabuleiro de Xadrez, típicas no processamento de imagens digitais, são baseadas na localização do pixel e não em suas intensidades. Portanto, o cálculo de suas respectivas distâncias intervalares, permanecem inalteradas (LYRA et al., 2004).

2.3.2.4 Operações entre pixels intervalares

As *operações aritméticas* entre pixels intervalares, conforme visto no capítulo 2 deste trabalho, são a soma, subtração, multiplicação e divisão.

As *operações lógicas* para imagens binárias intervalares, onde cada pixel possui exatamente a intensidade [0;0] ou [1;1] é análoga a usada para imagens digitais tradicionais. O resultado obtido também será similar ao do processamento tradicional, já que para a definição da imagem intervalar será mantida a posição no espaço para cada pixel intervalar, somente modificando a forma de apresentação para seus tons de cinza (agora definidos por intervalos), usados em tarefas como máscaras, detecção de características e análise em formas e em operações geradas pela vizinhança.

O processamento da vizinhança é realizado através de máscaras (filtros), que modificam o valor de um pixel intervalar em função de seu próprio nível de cinza e de seus vizinhos. A definição de operações lógicas, para imagens intervalares não binárias, analisam as intensidades dos pixels, não suas posições espaciais. São usadas técnicas para sobreposição de imagens, refinamentos, etc (LYRA et al., 2004).

- ✓ *Disjunção de imagens intervalares*

Sejam $A = a_{ij}$ e $B = b_{ij}$ imagens intervalares de ordem $m \times n$. A disjunção destas imagens é a matriz $C = c_{ij}$ de ordem $m \times n$, construída *pixel* a *pixel,* escolhendo aquele de intensidade maior. Isto é, $A \vee B = C$, onde $c_{ij} = \boldsymbol{sup}[a_{ij}, b_{ij}]$, para todo i, j (CRUZ, 2008; LYRA et al., 2004).

- ✓ *Conjunção de imagens intervalares*

Sejam $A = a_{ij}$ e $B = b_{ij}$ imagens intervalares de ordem $m \times n$. A conjunção destas imagens é a matriz $C = c_{ij}$ de ordem $m \times n$, construída *pixel* a *pixel,* escolhendo aquele de intensidade menor. Isto é, $A \wedge B = C$, onde $c_{ij} = \boldsymbol{inf}[a_{ij}, b_{ij}]$, para todo i, j (CRUZ, 2008; LYRA et al., 2004).

- ✓ *Negação de imagens intervalares*

Sejam $A = a_{ij}$ e $K = k_{ij}$ imagens intervalares de ordem $m \times n$. A negação de A é a matriz $\neg A = \neg a_{ij}$ de ordem $m x n,$ construída *pixel* a *pixel,* escolhendo o

complemento com respeito a K. Isto é, $\neg A = K - A$. Assim $\neg a_{ij} = k_{ij} - a_{ij}$, para todo i, j (CRUZ, 2008; LYRA et al., 2004).

2.3.3 Segmentação em imagens intervalares

Uma etapa importante em processamento de imagens digitais é a segmentação de imagens, que é o primeiro passo no processo de análise de imagens. Nesta, a imagem é subdvidida em partes ou objetos. A minimização ou controle de qualquer erro neste passo é fundamental para um melhor resultado da análise (TAKAHASHI; BEDREGAL; LYRA, 2004; 2005).

O uso da matemática intervalar associada ao processamento de imagens digitais tem como objetivo controlar possíveis erros computacionais. O processamento de imagens digitais intervalares é uma teoria recente, e Takahashi et al. (2004, 2005) desenvolveram um trabalho onde apresentam a segmentação de imagens intervalares, utilizando a técnica de agrupamento baseada na similaridade da intensidade luminosa das imagens, com o método k-means em versão intervalar.

O método k-means para segmentação de imagens particiona a imagem original em k grupos e atribui um valor médio que representará a intensidade luminosa (ou nível de cinza) referente a cada grupo (centróide), considerando imagens monocromáticas.

Este método busca diminuir a quantidade de níveis de cinza da imagem original para k quantidades, onde a média de valores dos elementos (ou pixels) de cada grupo representará uma intensidade luminosa, deixando a imagem somente com k intensidades luminosas diferentes, divididas em k agrupamentos para uma posterior análise dessa imagem segmentada.

Diversos algoritmos podem representar o método k-means, e abaixo segue uma destas representações, segundo Takahashi, Bedregal e Lyra (2004, 2005):

- ✓ ***Passo 1:*** entrar com a imagem, inicializar a quantidade k de grupos (clusters), inicializar os valores iniciais dos k centróides e definir o critério de parada;
- ✓ ***Passo 2:*** calcular a similaridade entre os pixels em cada centróide;
- ✓ ***Passo 3***: atribuir cada pixel com o grupo do centróide mais similar;
- ✓ ***Passo 4***: calcular a média de cada grupo e atribuir esse valor a cada centróide, de acordo com os grupos;
- ✓ ***Passo 5***: verificar o critério de parada, caso não seja satisfeito, voltar ao ***passo 2***.

Para o cálculo da medida de similaridade pode ser utilizada a distância Euclidiana entre o centróide e o pixel, e esta medida definirá a qual grupo esse pixel pertencerá. Em Takahashi, Bedregal e Lyra, (2005) as métricas intervalares utilizadas foram a métrica usual da matemática intervalar, a métrica de Moore e a Quasi-métrica.

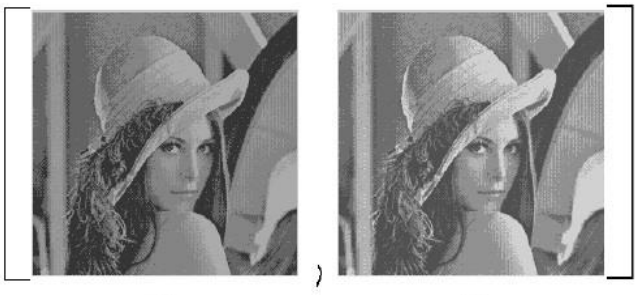
Sejam P = [p1, p2] e Q = [q1, q2] dois valores intervalares, então as métricas intervalares são:

- ✓ Métrica de Moore: $D_M = \max(|p1 - q1|, |p2 - q2|)$;
- ✓ Quasi-métrica: $D_{QM} = \max(q1 - p1, p2 - q2, 0)$.

O critério de parada utilizado no k-means intervalar foi a estabilização dos valores intervalares dos centróides, igual ao critério de parada do k-means tradicional, e uma quantidade máxima de épocas, ou seja, uma época é caracterizada pela obtenção do novo valor dos centróides intervalares durante a execução do método intervalar. Então, se os valores dos centróides não variarem, comparando os centróides atuais com os anteriores durante a execução do método, então o método encontrou um resultado que convergiu (TAKAHASHI; BEDREGAL; LYRA, 2004, 2005).

O método k-means intervalar para segmentação agrupa os pixels nos grupos de acordo com a similaridade dos centróides. Por exemplo, se um certo pixel possuir o valor intervalar de [75, 80], e se houver dois centróides cujos valores sejam: [80, 82] e [90, 95], então, o pixel [75, 80] será mais similar do centróide [80, 82] de acordo com as duas métricas intervalares.

A Figura 23 mostra uma imagem intervalar segmentada em oito grupos, ou seja, k=8, através do método k-means intervalar. Os itens (a) e (b) representam os ínfimos e os supremos, respectivamente, da imagem intervalar segmentada, ou seja, a imagem em (a) foi elaborada considerando os valores ínfimos dos intervalos de cada pixel da imagem, e (b) considerou os supremos (TAKAHASHI; BEDREGAL; LYRA, 2005).



(a) (b)

Figura 23 – Resultado da aplicação do método de segmentação intervalar k-means.
Fonte: Takahashi, Bedregal, Lyra, 2005, página 6.

Takahashi, Bedregal e Lyra (2005) concluíram seu artigo sobre o trabalho desenvolvido afirmando que nos testes realizados a segmentação com método k-means intervalar foi melhor que o método k-means tradicional, considerando o erro médio total encontrado nestes.

E, em relação a comparação entre as métricas intervalares utilizadas, consideraram que a métrica Quasi-métrica obteve um melhor desempenho, considerando o erro médio e desvio padrão obtido, embora tenha tido perdas quanto ao tempo de execução, ou seja, foi o mais lento para convergir ao resultado final.

Outra conclusão apontada por Takahashi, Bedregal e Lyra (2005) foi que quanto maior a quantidade de grupos menor o erro, ou seja, a segmentação por agrupamento obteve significativamente menos erros quanto maior o valor de k utilizado, tendo sido realizado testes com k=2, k=4 e k=8.

Outro trabalho com processamento de imagens intervalares encontrado na literatura foi o de Lorenz (2008), que se baseia na tranformada de Hough, usada em visão computacional, para reconhecimento de caixas em automação industrial. Em seu trabalho Lorenz aborda a versão da transformada de Hough, para reconhecimento de retas, associada a aritmética intervalar, quando considerou alcançar mais robustez. Em seu artigo citou resultados práticos com a implementação em C++ e concluiu ser um método poderoso para reconhecimento de padrões em imagens.

Neste capítulo foram abordados conceitos de imagens intervalares, técnicas de obtenção da mesma, morfologia matemática, exemplos de operações de erosão e de dilatação sobre a imagem intervalar, sua representação por matriz intervalar e segmentação, assuntos que serão pertinentes à mehor compreensão deste trabalho.

2.4 REDES NEURAIS ARTIFICIAIS

O homem busca entender como o pensamento é formado a milhares de anos e a área da inteligência artificial busca, além disso, construir entidades inteligentes, recriando esta capacidade de forma artificial (RUSSEL, NORVIG, 2004).

De acordo com BRAGA et al. (2000), Redes Neurais Artificiais são técnicas computacionais que apresentam um modelo matemático inspirado na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência.

Segundo HAYKIN (2001, p. 28), “uma rede neural é uma máquina que é projetada para modelar a maneira como o cérebro realiza uma tarefa particular ou função de interesse.“ Há toda uma classe de redes neurais artificiais que executam computação útil através de um processo de aprendizagem, ou seja, são capazes de generalizar uma experiência conhecida para demais ocorrências similares, obtendo assim o que se entende por conhecimento.

2.4.1 Neurônio

Em 1943 McCulloch e Pitts desenvolveram o primeiro neurônio artificial, ideia que veio a ter maior desenvolvimento na década de 80, quando surgiu mais intensamente as implementações de modelos de redes neurais artificiais.

Um neurônio pode ser visto como a menor unidade fundamental de processamento de uma rede neural e possui três elementos básicos em seu modelo: os pesos sinápticos, a junção aditiva e a função de ativação, conforme a figura 24 apresenta (HAYKIN, 2001).

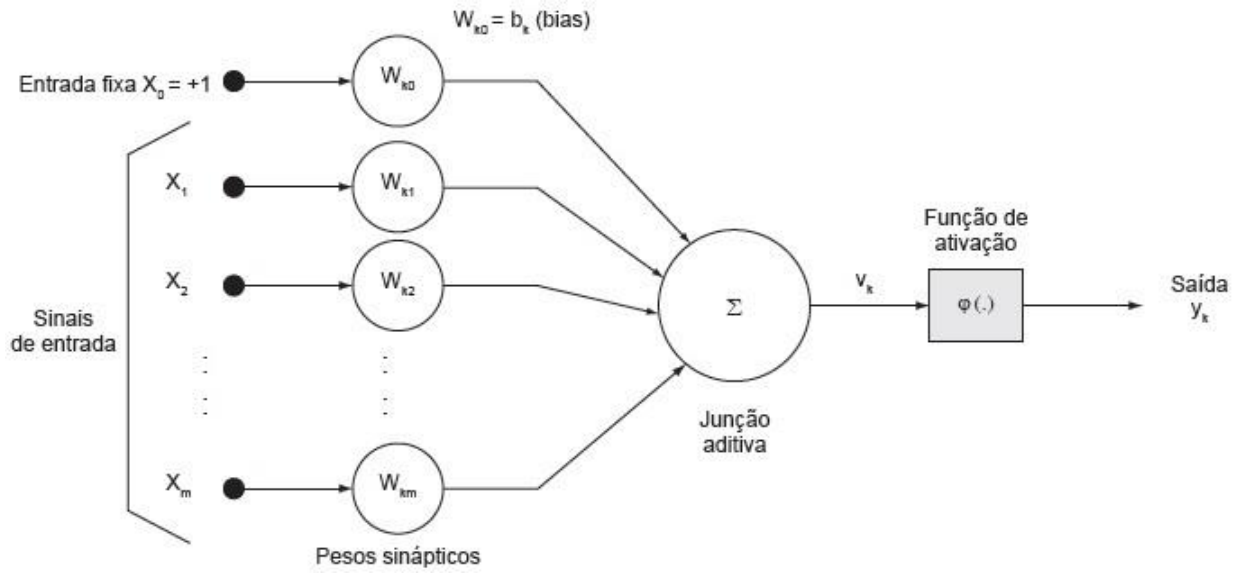


Figura 24 – Modelo não – linear de neurônio.
Fonte: Haykin, 2001.

Em termos computacionais, pode-se considerar que um neurônio é uma unidade de processamento de informação fundamental para uma rede neural (OTUYAMA et al., 2000).

Segundo HAYKIN (2001) pode-se descrever as partes que compõem o neurônio como segue:

- O conjunto de pesos sinápticos em um neurônio artificial podem incluir valores negativos, e é formado pelo valor das entradas X_j multiplicado por seu respectivo peso W_{kj}, onde *k* indica o neurônio e *j* a entrada.
- A junção aditiva faz a soma dos sinais de entrada e seus respectivos pesos, citados anteriormente, de forma ponderada pelas sinapses, funcionando como um combinador linear.
- E a função de ativação restringe o intervalo permissível de amplitude do sinal de saída, entre 0 e 1 ou -1 e 1.

No modelo apresentado de neurônio na figura 24 tem-se a entrada bias aplicada externamente, representada por b_K, que aumenta ou diminui a entrada na função de ativação.

Em termos matemáticos Haykin nos apresenta o par de fórmulas para descrição de um neurônio k, ou seja, u_k e y_k nas equações 12 e 13.

$$u_k = \sum_{j=1}^{m} w_{kj} x_j \tag{12}$$

$$y_k = \varphi(u_k + b_k) \quad (13)$$

Onde $x_1, x_2, \ldots, x_m$ representam os sinais de entrada; $w_{k1}, w_{k2}, \ldots, w_{km}$ são os pesos sinápticos do neurônio k; u_k é a saída do combinador linear dos sinais de entrada e dos pesos sinápticos; b_k é o bias, que aplica uma transformação *afim* à saída u_k; $\varphi(.)$ é a função de ativação e y_k é o sinal de saída do neurônio.

O potencial de ativação ou campo local induzido v_k é entendido conforme equação 14:

$$v_k = u_k + b_k \quad (14)$$

2.4.1.1 Função de Ativação

Pode-se considerar três tipos básicos de função de ativação: a função de limiar, a função linear por partes e a função sigmóide, definidas conforme o campo induzido v (HAYKIN, 2001).

Em função de limiar tem-se que:

$$y_k = \begin{cases} 1 \; se \; v_k \geq 0 \\ 0 \; se \; v_k < 0 \end{cases}$$

Descreve a propriedade “tudo ou nada” do modelo de neurônio de McCulloch-Pitts.

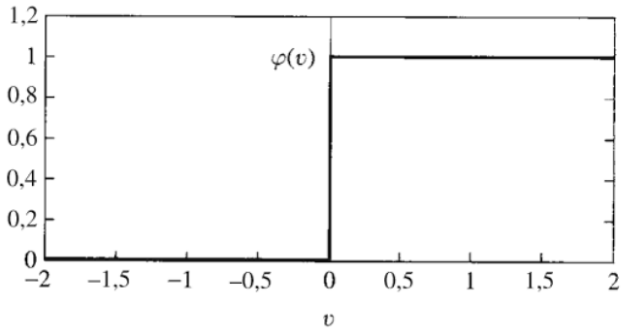


Figura 25 – Função de ativação de limiar
Fonte: Haykin, 2001

Em função linear por partes tem-se que:

$$\varphi(v) = \begin{cases} 1, & v \geq +1/2 \\ v, & +\frac{1}{2} > v > -1/2 \\ 0, & v \leq -1/2 \end{cases}$$

Esta forma é vista como uma aproximação de um amplificador não-linear.

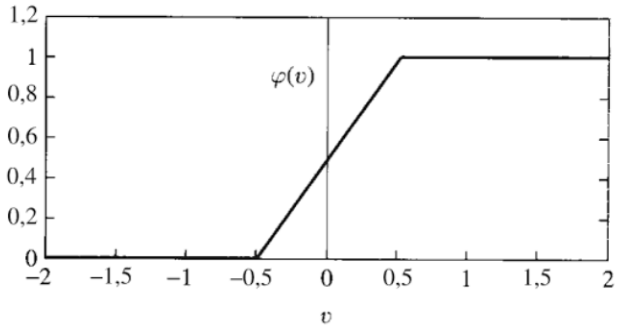


Figura 26 – Função de ativação linear por partes
Fonte: Haykin, 2001

Em função sigmóide tem-se que:

$$\varphi(v) = \tanh(v)$$

A função de ativação utiliza a função tangente hiberbólica, e esta é a forma mais comum das funções de ativação utilizadas em redes neurais artificiais. Define uma função estritamente crescente que exibe um balanceamento adequado entre comportamento linear e não-linear.

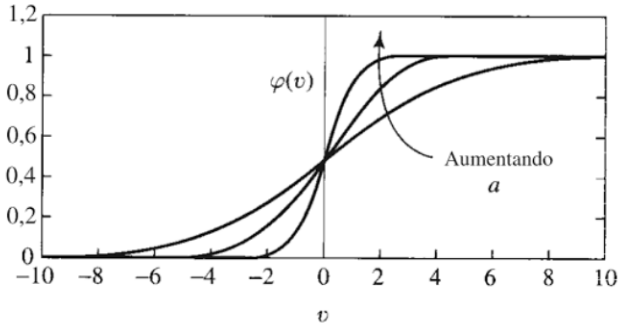


Figura 27 – Função de ativação sigmóide
Fonte: Haykin, 2001

2.4.2 Aprendizagem

Segundo Haykin (2001): “A propriedade que é de importância primordial para uma rede neural é a sua habilidade de aprender a partir de seu ambiente e de melhorar o seu desempenho através da aprendizagem”.

Aprender é o ato que produz um comportamento diferente a um estímulo externo devido a excitações recebidas no passado e é de uma certa forma sinônimo de aquisição de conhecimento (BARRETO, 2002).

As redes neurais possuem a capacidade de aprenderem por exemplos e depois serem capazes de generalizar, predizendo saídas a entradas ainda não

testadas. O aprender, o conhecimento adquirido, por assim dizer, está armazenado nos valores das conexões sinápticas.

> Aprendizagem é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estimulação pelo ambiente no qual a rede está inserida. O tipo de aprendizagem é determinado pela maneira pela qual a modificação dos parâmetros ocorre (HAYKIN, 2001, p. 75).

Uma aprendizagem pode ser considerada supervisionada quando ela é realizada com um professor, entendendo-se que o mesmo possui conhecimento sobre o ambiente e representa um conjunto de exemplos de entrada e saída. Tem-se um vetor de treinamento onde o professor informa à rede neural a resposta para a entrada recebida e os parâmetros da rede são ajustados de acordo com a resposta esperada e o sinal de erro. Esse ajuste é realizado passo a passo, objetivando que a rede neural emule ao professor, quando a rede neural assimilaria o conhecimento deste. Quando esta condição é alcançada a rede pode dispensar o mesmo e lidar com o ambiente por conta própria, saindo da fase considerada treinamento e entrando na chamada fase de testes. É um tipo de aprendizagem por correção de erro (HAYKIN, 2001).

Uma aprendizagem sem um professor ou não supervisionada ocorre quando não há uma indicação expressa da saída adequada, ou seja, não há exemplos rotulados, como corretos ou não, da função a ser aprendida pela rede. Pode se dar pelo aprendizado por reforço, onde o aprendizado é realizado através de interação contínua com o ambiente buscando desenvolver a habilidade de aprender a realizar uma tarefa predeterminada, com base apenas no resultado de sua experiência resultante desta relação (HAYKIN, 2001).

2.4.3 Aplicações

As redes neurais são geralmente utilizadas na solução de problemas complexos e em sua execução podem exigir alto dispêndio de tempo, processamento e consumo de memória, especialmente no período de treinamento, podendo levar horas ou dias, conforme o caso (SOUZA, 2012).

Podem ser aplicadas em diversas áreas, como no processamento de imagens; previsão e reconhecimento de padrões; otimização de sistemas; aproximador universal de funções; controle de processos em robótica, aeronaves

ou satélites; agrupamento de dados (clusterização); memória associativa; prognósticos no mercado financeiro; etc (SOUZA, 2012).

2.4.4 Tipos de arquiteturas

Segundo HAYKIN (2001), podemos identificar três tipos de arquiteturas fundamentalmente diferentes: Redes alimentadas adiante com camada única, Redes alimentadas diretamente com múltiplas camadas e redes recorrentes.

As redes alimentadas adiante com camada única são a forma mais simples de organização, onde tem-se a camada de nós de entrada e a camada de nós de saída e as informações da camada de entrada são direcionadas à saída e não vice-versa, possuindo uma única direção: adiante. Apenas a camada de saída faz computação, por isso chamada de rede de camada única.

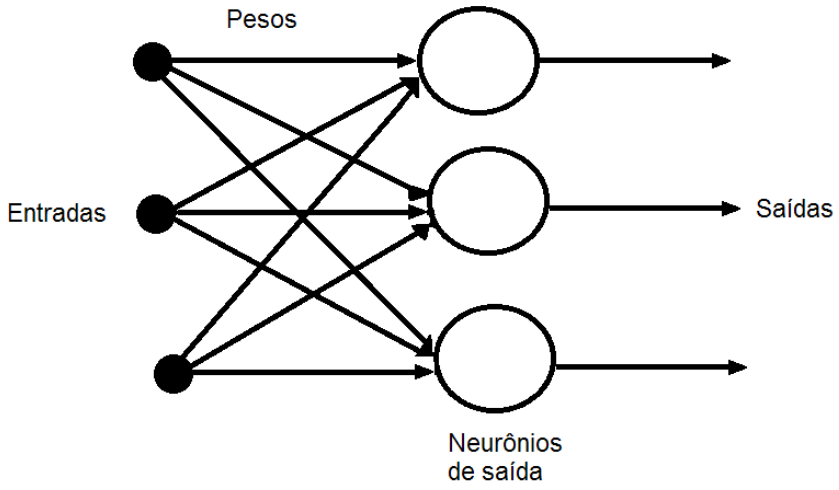


Figura 28 - Modelo de rede alimentada adiante com camada única
Fonte: Haykin, 2001.

As redes alimentadas diretamente com múltiplas camadas se distinguem por possuir uma ou mais camadas ocultas de neurônios realizando processamento. Com a adição de camadas ocultas a rede torna-se capaz de extrair estatísticas de ordem elevada, adquirindo uma perspectiva global apesar de sua conectividade local. A informação que percorre a rede também possui apenas a direção de adiante na rede.

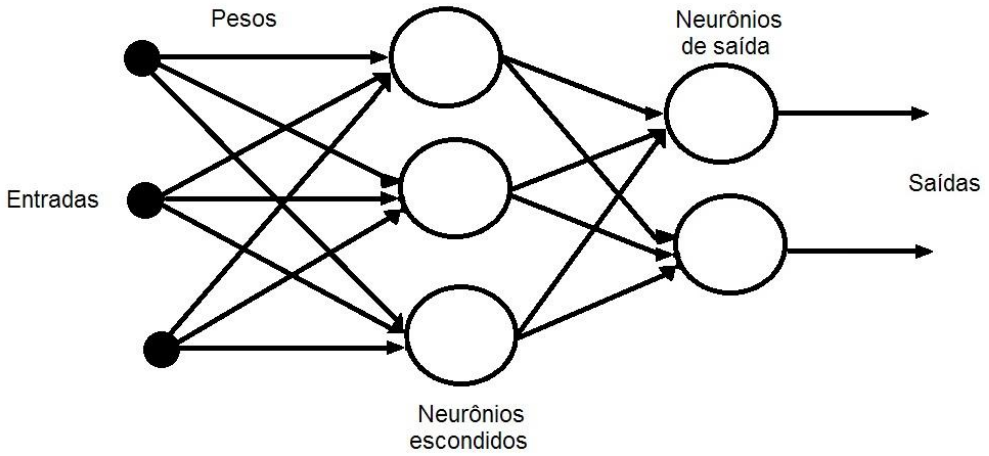


Figura 29 - Modelo de rede alimentada adiante de múltiplas camadas

Fonte: Haykin, 2001.

As redes recorrentes se distinguem por possuir ao menos um laço de realimentação, ou seja, existirá alguma informação que será retropropagada na rede. Pode observar-se não haver auto-realimentação, pois a saída de um mesmo neurônio não é conectada a sua própria entrada. A existência de laços de realimentação em redes neurais artificiais tem um impacto profundo na sua capacidade de aprendizagem e em seu desempenho.

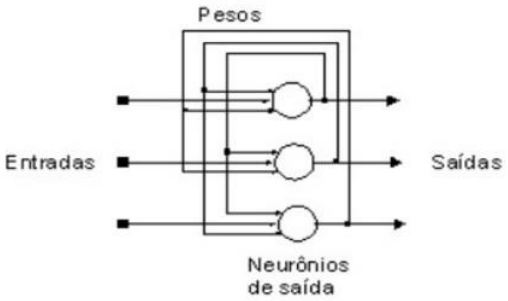


Figura 30 - Modelo de rede recorrente
Fonte: Haykin, 2001.

Uma rede neural é dita totalmente conectada quando todos os nodos ou neurônios de uma camada são conectados com todos os nós da camada seguinte. Mas se houverem algumas conexões sinápticas faltando, ela será parcialmente conectada (HAYKIN, 2001).

As redes podem ser agrupadas de diferentes formas, como pela quantidade de camadas, tipo de aprendizado utilizado, conexões entre neurônios e podemos citar diversas arquiteturas conhecidas como: Perceptron, Adaline, Hopfield, Kohonen, Múltiplas Camadas Percetron, Convolutiva, entre outras. Neste trabalho serão utilizadas as arquiteturas de MLP (Múltiplas Camadas

Perceptron ou *MultiLayer Perceptron*) e CNN (Rede neural Convolutiva ou *Convolutional Neural Network*).

2.4.4.1 Rede Perceptron

Primeiro modelo de rede neural implementado em 1958 por Rosenblatt, quando demonstrou que as sinapses poderiam ser ajustadas e treinadas para classificar certos tipos de padrões. Ele descreveu a estrutura de ligação entre neurônios e propôs um algoritmo de treinamento para esta topologia de rede (BRAGA et al., 2000).

A rede perceptron possui três camadas, a primeira recebe os sinais de entrada por conexões fixas, a segunda recebe impulsos da primeira através de conexões cujos pesos são ajustáveis e envia as respostas de saída para a terceira camada (BRAGA et al., 2000).

Este tipo elementar de rede, com única camada de neurônios efetuando processamento, comporta-se como um classificador de padrões linearmente separável, ou seja, classifica padrões que estão em lados opostos de um hiperplano. Inicialmente a saída da rede é aleatória, mas com o ajuste gradual dos pesos ela é treinada para fornecer saídas de acordo com os dados do conjunto de treinamento (BRAGA et al., 2000).

Segundo HAYKIN (2001): “o perceptron é a forma mais simples de uma rede neural usada para a classificação de padrões ditos linearmente separáveis.”

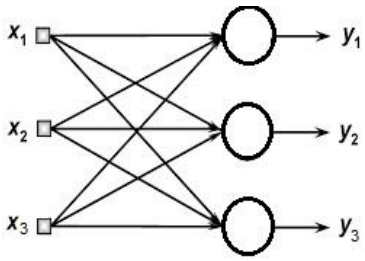


Figura 31 - Rede de camada única perceptron
Fonte: Haykin, 2001.

2.4.4.2 Perceptron de Múltiplas Camadas

A rede perceptron de múltiplas camadas ou MLP de *MultiLayerPerceptron* são redes alimentadas adiante, com nós ou neurônios distribuídos em camadas, constituindo a camada de entrada, uma ou mais camadas ocultas e a camada de

saída. São aplicadas com sucesso na resolução de diversos problemas difíceis, com treinamento supervisionado com um algoritmo de retropropagação, ou *back-propagation*, considerado muito popular (HAYKIN, 2001).

O algoritmo de retropropagação é baseado na regra de aprendizagem por correção de erro, que consiste basicamente em dois passos através das camadas de rede: a propagação do sinal de entrada aplicado aos neurônios (ou nós computacionais) fluindo através da rede e a retropropagação do sinal de erro, que flui em sentido contrário, para trás na rede (HAYKIN, 2001).

Durante o passo para frente, na propagação do sinal, os pesos sinápticos são todos fixos. Já no passo para trás, na retropropagação pela rede, os pesos sinápticos são reajustados de acordo com uma regra de correção e erro. A resposta real da rede é subtraída de uma resposta desejada para produzir um sinal de erro. Isto ocorre na fase de treinamento da rede. Segundo HAYKIN (2001, p.184): ”Os pesos sinápticos são ajustados para fazer com que a resposta real da rede se mova para mais perto da resposta desejada, em um sentido estatístico”.

Haykin considera que uma rede perceptron de múltiplas camadas possui três características distintivas:

- O modelo de cada neurônio da rede inclui uma função de ativação não-linear, com não linearidade suave, ou seja, diferenciável em qualquer ponto. Uma forma comumente utilizada é a sigmóide definida pela função da equação 15:

$$y_k = \frac{1}{1+ e^{(-v_k)}} \tag{15}$$

  onde v_k é a soma ponderada de todas as entradas sinápticas acrescida do bias do neurônio *k* e y_k é a saída deste neurônio.
- A rede possui uma ou mais camadas de neurônios ocultos, que não são parte da camada de entrada ou de saída. Estes capacitam a rede no aprendizado de tarefas complexas por extrair progressivamente as características mais significativas dos padrões de entrada.
- A rede neural de múltiplas camadas possui um alto grau de conectividade, determinada pelas sinapses da rede.

Segundo HAYKIN (2009) é devido a combinação destas características e da habilidade de aprender com a experiência, que o perceptron de múltiplas camadas

obtêm seu poder computacional, tornando difícil a sua análise teórica. Também considera mais difícil o processo de aprendizagem ser visualizado, com a utilização de neurônios ocultos.

Redes com dois ou mais neurônios nas camadas intermediárias (ocultas) podem ser utilizadas para a classificação de problemas não lineares (BRAGA et al. 2000).

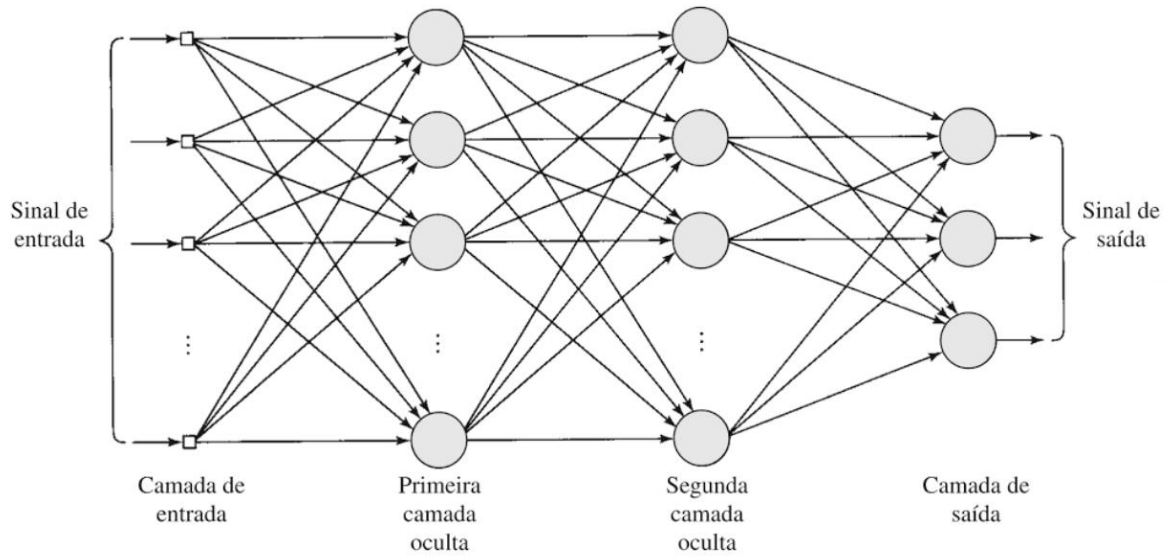


Figura 32 -Grafo de um Perceptron Múltiplas Camadas com duas camadas ocultas
Fonte: Haykin, 2001.

Haykin (2001) refere-se a dois sinais básicos dos Perceptrons de Múltiplas Camadas com aprendizagem por retropropagação: Os sinais funcionais e os sinais de erro.

Um sinal funcional é um estímulo recebido na camada de entrada que se propaga para a frente, neurônio por neurônio, através da rede e emerge na camada de saída como um sinal de saída. Este sinal é calculado como uma função não linear de suas entradas e pesos associados, aplicados ao respectivo neurônio intermediário ou de saída. Também chamado de sinal de entrada.

Sinais de erro se originam em um neurônio de saída da rede e se propagam no sentido inverso, de trás para frente na rede, camada por camada. O cálculo de seu valor envolve uma função dependente do erro, como uma estimativa do vetor gradiente para a retropropagação na rede.

O sinal de erro do neurônio de saída k, na iteração *n*, ou seja, no *n-ésimo* exemplo de treinamento, é definido por (HAYKIN, 2001) conforme equação 16:

$$e_k(n) = d_k(n) - y_k(n), \tag{16}$$

onde $e_k(n)$ se refere ao sinal de erro na saída do neurônio *k*, na iteração *n*; $d_k(n)$ se refere a resposta desejada na saída do neurônio *k*, na iteração *n*; e $y_k(n)$ se refere ao sinal funcional que aparece na saída do neurônio *k*, na iteração *n.*

Haykin (2001) definiu o valor instantâneo da energia do erro para o neurônio k como: $\frac{1}{2}e_k^2(n)$

e correspondentemente a energia total do erro é obtida somando-se o erro instantâneo do erro de todos os neurônios da camada de saída, conforme equação 17:

$$E(n) = \frac{1}{2}\sum_{k \in C} e_k^2(n), \quad (17)$$

onde o conjunto C inclui todos os neurônios da camada de saída da rede.

Considerando *N* como o número total de exemplos de padrões contidos no conjunto de treinamento, a energia média do erro quadrado é obtida somando-se os erros totais para todos os *n* exemplos e então normalizando em relação ao tamanho do conjunto N, pela equação 18:

$$E_{med} = \frac{1}{N}\sum_{n=1}^{N} E(n) \quad (18)$$

O desempenho do treinamento é medido pelo erro médio quadrado definido como a média dos erros instantâneos, para todos os N vetores de treinamento.

A energia instantânea do erro *E(n)* e a energia média do erro E_{med} são funções de todos os parâmetros livres da rede, isto é, os pesos sinápticos e níveis de bias. O objetivo do processo de aprendizagem é ajustar os parâmetros livres da rede para minimizar o E_{med} (HAYKIN, 2001).

Considerou-se um método simples de treinamento onde os pesos são alterados “padrão a padrão” até formarem uma época, ou seja, uma apresentação completa do conjunto de treinamento que está sendo processado. “Os ajustes de pesos são realizados de acordo com os respectivos erros calculados para cada padrão apresentado à rede” (HAYKIN, 2001, p.188).

No algoritmo de retropropagação ou *backpropagation* tem-se a taxa de aprendizado, uma constante de proporcionalidade que define a rapidez com que o vetor de pesos é modificado na busca do ajuste de pesos (BRAGA et al., 2000).

Com a abordagem de treinamento por padrão, quando os pesos são atualizados após cada apresentação do padrão de treinamento, a estabilidade

ocorre quando a taxa de aprendizado for pequena. Quando taxas de aprendizado elevadas são utilizadas a rede geralmente se torna instável (BRAGA et al., 2000).

2.4.4.3 Redes Neurais Convolucionais

As redes neurais convolucionais, convolutivas ou CNN, de *Convolutional Neural Networks*, foram inicialmente sugeridas em 1998 por LeCun, Bottou, Bengio e Haffner e são arquiteturas que podem ser treinadas e aprenderem representações invariantes a escalas, translação, rotação e transformações afins (JURASZEK, 2014 apud LECUN et al., 2010).

As redes convolutivas são um tipo particular de redes neurais artificiais, que são do tipo *feedforward*[1]. Além disso, este tipo de rede possui conexões locais, pesos compartilhados e uso de várias camadas, sua estrutura é uma série de estágios de duas camadas: convolução, gerando *featuremaps* e *pooling* (LECUN et al., 2015)

Esta rede se baseia no uso de células simples e complexas para extrair características implícitas dos padrões visuais apresentados como entrada, e que são integradas a uma rede completamente conectada, a qual realiza a classificação de padrões a partir de características extraídas pela última camada de células complexas (FERNANDES, 2013).

Redes neurais convolucionais são projetadas para processar dados em forma de múltiplos *arrays*, como, por exemplo, em imagens bidimensionais. Possuem quatro idéias centrais que se aproveitam de propriedades naturais de sinais: conexões locais, pesos compartilhados, *pooling* e o uso de muitas camadas. A sua arquitetura típica é estruturada em uma série de estágios. O primeiro conjunto de estágios é composto por dois tipos de camadas: camadas de convolução e de *pooling* ou subamostragens (LECUN et al., 2015).

Unidades em uma camada convolucional são organizados em mapas de características ou *featuremaps,* que estão conectados com fragmentos locais dos mapas de características da camada anterior, através de um conjunto de pesos chamados de banco de filtros. Diferentes mapas de características na camada usam

[1] Em uma rede *feedforward,* cada camada se conecta à próxima camada, porém não há caminho de volta.

diferentes bancos de filtros. A razão para esta arquitetura é dupla (LECUN et al., 2015):

Primeiro, em um vetor de dados como as imagens os grupos locais de valores são muitas vezes altamente correlacionados, formando temas locais distintos que são facilmente detectados.

Segundo, estatísticas locais de imagens e outros sinais são invariantes para localização, ou seja, se um tema pode aparecer em uma parte da imagem, ele pode aparecer em qualquer lugar, consequentemente surge a ideia de unidades de localização diferentes compartilhando o mesmo peso e detectando o mesmo padrão em diferentes partes do vetor. Matematicamente operações de filtragem realizadas em mapas de características são convoluções discretas (LECUN et al., 2015).

“Convolução é o processo que calcula a intensidade do pixel em função da intensidade dos seus vizinhos” (TORTELLI, 2016, p. 43).

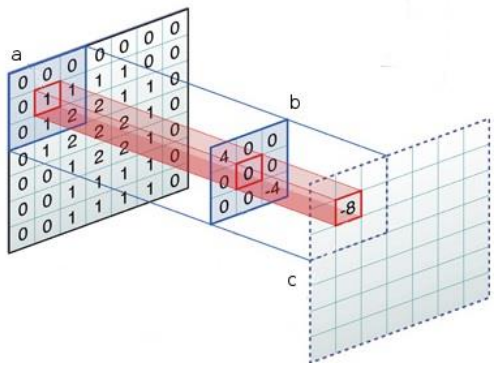


Figura 33 – Representação de uma convolução aplicada a somente uma região de entrada (a) imagem ou mapa de característica (b) filtro

A camada de *pooling* ou subamostragem que fica entre as camadas de convolução faz a diminuição do tamanho de representação espacial da imagem de entrada ou dos mapas de características (TORTELLI, 2016).

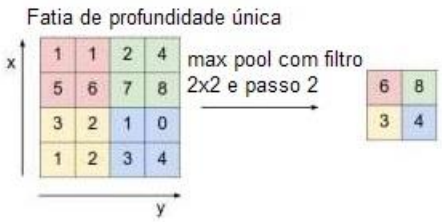


Figura 34 – Exemplo de *max pooling* de tamanho 2x2 aplicado em uma entrada bidimensional 4x4, onde se propaga apenas o valor mais elevado.
Fonte: Tortelli, 2016.

Apesar do papel da camada de convolução ser detectar conjunções locais das camadas de características da camada anterior, a função da camada de *pooling* é mesclar características semanticamente similares em uma. Após as camadas de convolução e *pooling* segue a camada totalmente conectada (LECUN et al., 2015).

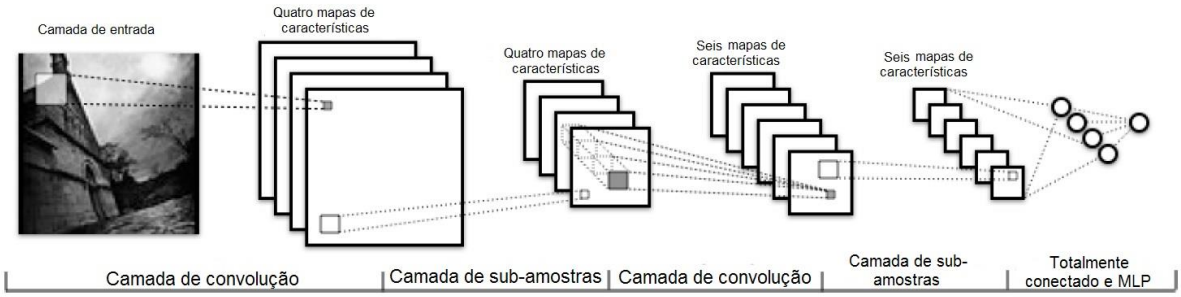


Figura 35 – Modelo de arquitetura de rede neural convolucional
Fonte: Tortelli, 2016.

Redes neurais convolucionais foram usadas em numerosas aplicações como, por exemplo, em reconhecimento de discurso, reconhecimento de faces, em compreensão de linguagem natural e sistemas de visão para carro. Revolucionaram a área de *Machine Learn* (Aprendizado de Máquina). Desde 2000, estas redes têm sido aplicadas com grande sucesso para a detecção, segmentação e reconhecimento de objetos e regiões em imagens. Elas são agora as abordagens dominantes para quase todas as tarefas de reconhecimento e de detecção e abordagens de desempenho humano em algumas tarefas (LECUN et al., 2015).

Neste capítulo foram apresentadas concepções de redes neurais artificiais, neurônios, aprendizagem e aplicações. Apresentou também algumas arquiteturas, como de redes neurais convolucionais, tópico significativo à este trabalho.

3 REDES NEURAIS CONVOLUCIONAIS INTERVALARES COM IMAGENS INTERVALARES

Este trabalho propõe o uso de imagens intervalares (LYRA, 2003) como entradas em uma extensão de rede neural convolucional intervalar. Tem como base o trabalho de TORTELLI (2016) em que se tem uma rede neural convolucional intervalar com propósito de controlar e automatizar a análise do erro numérico, onde as camadas que compõem a rede neural por convolução são representadas por operações equivalentes através de intervalos. Busca-se manter uma parametrização com este trabalho com a finalidade de comparação e tem-se por objetivo analisar se houve melhora na precisão e na classificação. O uso das imagens intervalares visa melhorar o processamento interno da rede, garantindo exatidão numérica e controle rigoroso do erro e assim os resultados provindos das camadas de convolução tendem a serem mais exatos até chegarem à entrada da MLP.

Em cálculos de ponto flutuante que ocorrem na computação convencional ocorrem erros de arredondamento e truncamento. Estes erros são propagados em operações computacionais no processamento de imagens nas redes convolucionais e a aritmética intervalar proposta por MOORE (1966) é usada de forma a prover um controle rigoroso do erro numérico, fornecendo assim resultados com maior confiabilidade. Este pode ocorrer desde a obtenção da imagem, na discretização dos valores, bem como nos cálculos que são envolvidos no seu processamento, como na convolução.

Com a utilização da aritmética intervalar os valores não são aproximados, sendo que o valor real x é representado em um intervalo X que o contenha (MOORE, 1966).

3.1 Imagem intervalar

A imagem intervalar é adquirida através do processamento de imagens convencionais, visto que não há equipamentos que as possam fornecer diretamente. O método utilizado neste trabalho se baseia na vizinhança de 8 (N_8) e na vizinhança de 4 (N_4) dos pixels, como LYRA (2003) sugeriu em seu trabalho, porém com diferenças substanciais de aplicação, visto termos uma proposta própria para o cálculo do novo pixel da imagem, este sendo intervalar.

Lembrando que uma imagem digital intervalar é uma matriz A de ordem m_xn que representa uma imagem discretizada e digitalizada em forma intervalar, onde o

valor de cada pixel representa um intervalo de tons de cinza, possuindo um valor inferior e superior.

Nossa proposta para a obtenção de uma imagem intervalar é calcular as distâncias entre o pixel e sua vizinhança, de 8 ou de 4, e obter o novo pixel intervalar conforme a equação 19.

$$p' = \left[p - \frac{d}{2}; p + \frac{d}{2}\right] \quad (19)$$

onde p é a representação do pixel, d consiste no menor diâmetro da vizinhança analisada e p' é o pixel intervalar obtido.

Em caso de todos os vizinhos possuírem a mesma distância do pixel observado e esta for nula, tem-se o pixel intervalar pontual ou degenerado, ou seja, um intervalo com diâmetro nulo, onde os valores inferiores e superiores são iguais.

Considerando a figura 36 como os valores de pixels de parte de uma imagem tradicional observa-se a menor distância entre os valores do pixel e sua vizinhança. Neste exemplo observa-se que a menor distância é 10 (a). Em (b) pode-se observar que o intervalo do novo pixel possui diâmetro de 10, sendo 5 abaixo do valor original do pixel e 5 acima. O antigo pixel de valor 100 tornou-se um novo pixel intervalar igual a [95, 105], em função da vizinhança de 8.

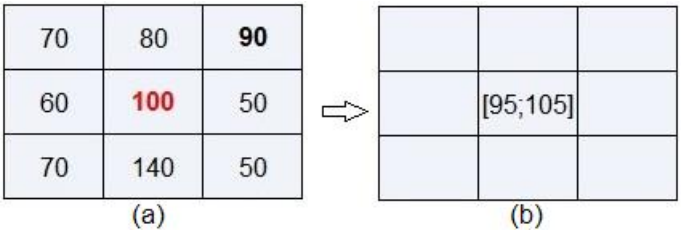


Figura 36 – Representação de pixel e seus vizinhos N8 e o pixel intervalar

E na figura 37 pode se observar o resultado desta mesma operação considerando a vizinhança de 4.

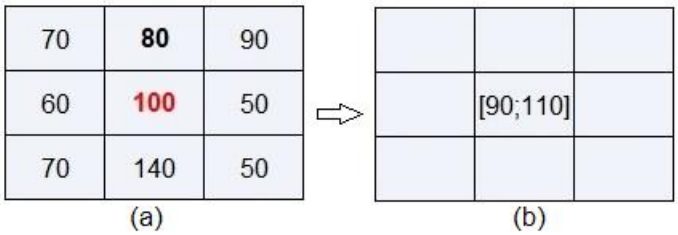


Figura 37 – Representação de pixel e seus vizinhos N4 e o pixel intervalar

Este cálculo é feito em todos os pixels da imagem convencional original, gerando uma nova imagem intervalar. Os pixels de borda consideram apenas os

pixels vizinhos existentes. Note-se que o diâmetro do novo pixel dependerá dos valores dos pixels da vizinhança considerada.

Da mesma maneira é realizado o cálculo considerando a vizinhança de 4 (N4) dos pixels.

A figura 38 e figura 39 apresentam a mudança na representação de uma imagem convencional e seus valores de pixels para valores de pixels intervalares. O valor original convencional do pixel fica encapsulado entre os valores inferior e superior do intervalo do novo pixel intervalar.

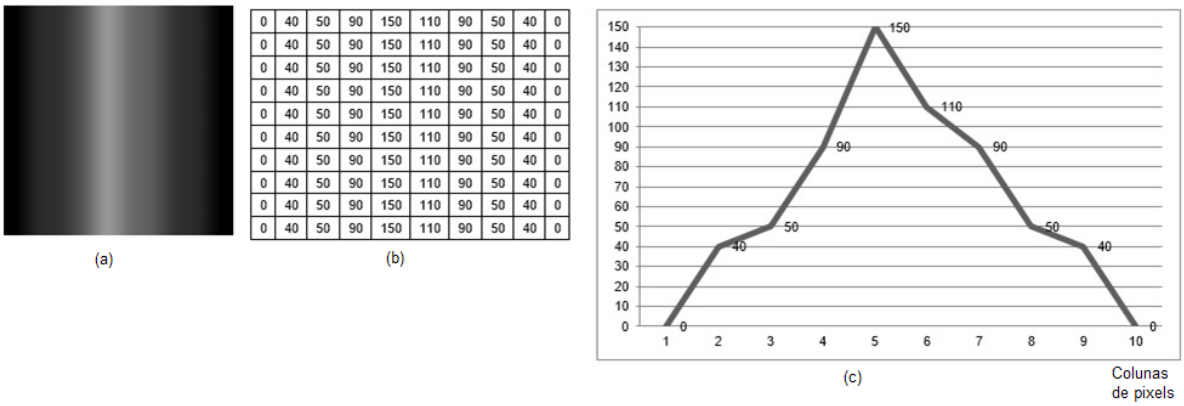


Figura 38 – Imagem convencional (a), valores de pixels (b) e gráfico tons de cinza (c)

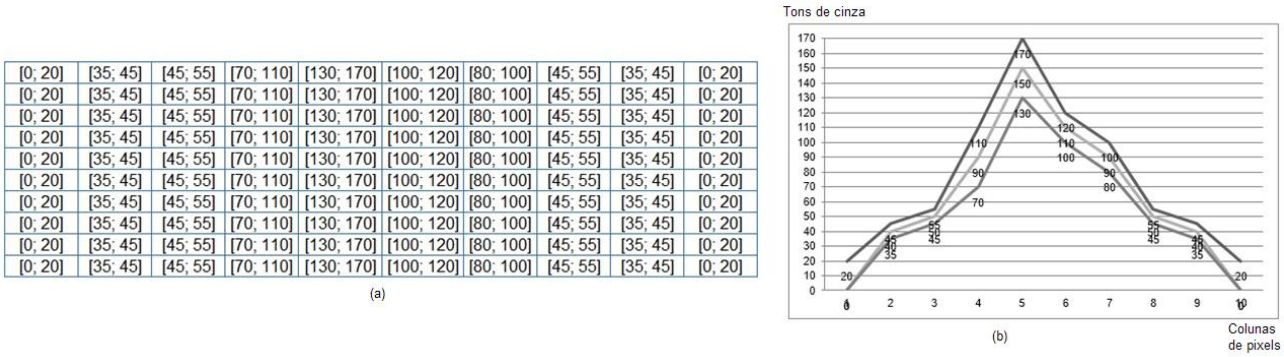


Figura 39– Visualização dos pixels intervalares equivalentes (a) e gráfico tons de cinza (b)

As figuras 38 e 39 demonstram bem o poder de encapsulamento que a matemática intervalar possui, aplicando se a imagens. O poder de representação da imagem intervalar é alto, embora não tenha uma aparente aplicabilidade prática de visualização, visto não dispormos de equipamentos que as possam exibir em sua forma natural.

3.2 Sugestão de visualização da imagem intervalar

A imagem intervalar não foi desenvolvida com foco em sua visualização. A sugestão de visualização no trabalho de Lyra e Alan Ribeiro (2003) era de montar uma imagem com os valores inferiores do intervalo e outra com os valores superiores. Nota-se que as duas imagens geradas são extremamente similares, a menos que o intervalo seja de grande diâmetro.

Neste trabalho sugere-se, caso se faça interessante uma visualização da imagem intervalar, uma forma alternativa com aparente maior significado. Uma única imagem seria visualizada a partir de uma única imagem intervalar. Seria observado o tamanho do intervalo em cada pixel e ao maior diâmetro seria atribuída a cor branca e ao menor diâmetro (intervalo degenerado) a cor preta e para os tamanhos intermediários de diâmetro seria usada uma escala de cinza. Com isso, teria-se a visualização da imagem intervalar de forma significativa, onde as áreas mais claras representariam os pixels de maior diâmetro e as áreas mais escuras, os pixels de menor diâmetro. Ou seja, nas bordas dos objetos a cor branca prevaleceria, pois é onde os maiores intervalos estariam. A figura 40 apresenta um exemplo desta sugestão de visualização.

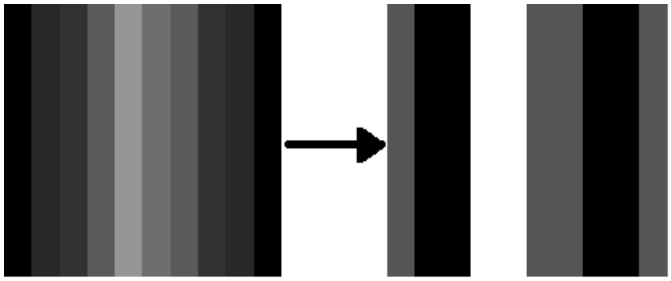


Figura 40 – Sugestão de visualização, imagem original (a) e imagem intervalar equivalente (b)

3.3 Configurações da rede convolucional intervalar

A rede proposta é composta por duas camadas de convolução, intermediadas por uma camada de *pooling* e seguidas por uma camada totalmente conectada, que compatibiliza os dados, adequando os mapas de características intervalares gerados e sua inserção em MLP ou Rede Perceptron Multicamadas convencional.

Nas camadas de convolução utiliza-se um filtro randômico intervalar, de tamanho 3x3. A convolução com este filtro gerará *feature maps* ou mapas de características bidimensionais que serão utilizados nas etapas seguintes. A

representação intervalar do filtro *w* é dada por $w = [w_1; w_2]$, sendo w_1 o limite inferior do intervalo e w_2 o limite superior.

O filtro 3x3, com cada elemento intervalar, pode ser representado por:

$$f(x,y) = \begin{bmatrix} [w_1(1,1);\ w_2(1,1)] & [w_1(2,1);\ w_2(2,1)] & [w_1(3,1);\ w_2(3,1)] \\ [w_1(1,2);\ w_2(1,2)] & [w_1(2,2);\ w_2(2,2)] & [w_1(3,2);\ w_2(3,2)] \\ [w_1(1,3);\ w_2(1,3)] & [w_1(2,3);\ w_2(2,3)] & [w_1(3,3);\ w_2(3,3)] \end{bmatrix}$$

A extensão intervalar de toda a camada de convolução com imagens intervalares pode ser dada pela expressão da equação 20:

$$X_{ij}^{l} = \sum_{a}^{m-1} \sum_{b}^{m-1} w_{ab} \Im_{(i+a)(j+b)}^{l-1} \tag{20}$$

Onde *X* é o mapa de características intervalares gerado na operação de convolução, *W* é o núcleo ou filtro de convolução, $\Im$ a imagem intervalar de entrada, *a* e *b* são as dimensões espaciais do mapa de características ou *feature map, i* e *j* são as dimensões da entrada $\Im$ e *m* consiste na dimensão generalizada do núcleo de convolução *W* e *l* indica o *feature map* gerado no momento.

Então X^l é o mapa de características intervalar gerado a partir da ponderação entre o filtro intervalar *W* e a entrada intervalar $\Im$.

A camada de *Pooling* visa diminuir a dimensão espacial de entrada, para uma execução em menor tempo. Apesar de ocorrer perda de informação, esta difundirá a característica de maior interesse, deixando um *feature map* de menor tamanho e com maior concentração de informações pertinentes. É utilizado o *Max Pooling*, método que seleciona o pixel de maior valor de uma região, estabelecida pela dimensão do *pooling*, para permanecer na nova configuração de imagem redimensionada. Esta camada também opera com o tipo intervalar. Tem-se a equação 21:

$$X_{ij} = \max \{ X_{i'j'} : i \leq i' < i + k,\ j \leq j' < j + k \} \tag{21}$$

onde (i, j) refere-se as coordenadas do novo pixel criado após o resultado do *pooling,* (i', j') refere-se as coordenadas do pixel de entrada na etapa de *pooling* e k indica a dimensão (k; k) do *pooling*, neste caso k = 2.

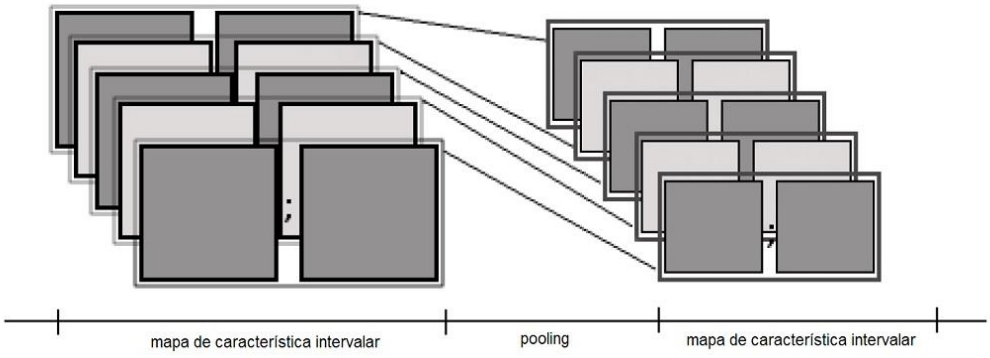


Figura 41 - Representação da camada de pooling intervalar.
Fonte: Tortelli, 2016.

A camada totalmente conectada consolida as informações dos mapas de características ou *feature maps* para inserção na MLP e subsequente classificação de dados. Esta camada deve manter a compatibilidade dos dados intervalares com a entrada não intervalar da MLP. A extensão intervalar foi utilizada neste trabalho, nos processos que agem sobre a imagem intervalar de entrada e em suas *feature maps* provenientes da convolução.

A MLP utilizada segue o modelo da aritmética real convencional, conforme trabalho realizado por TORTELLI (2016), e a camada totalmente conectada tem a tarefa de adequar os valores dos pixels intervalares dos *feature maps* para pixels convencionais. Até o momento não se tem disponibilidade de bibliotecas com MLPs intervalares.

Para compatibilizar os dados provenientes das camadas de convolução, os mapas de características intervalares com a entrada da MLP, é necessário converter estes dados para números de ponto flutuante. Para isso, utiliza-se o conceito de ponto médio, calculando-se o valor central de cada intervalo dos pixels dos mapas de características e utilizando este valor para formar os novos pixels dos referidos mapas, que serão inseridos na MLP que trabalha com a aritmética convencional.

O cálculo do ponto médio intervalar se dá pela equação 22:

$$x_{ij}^{l} = m\left(X_{ij}^{l}\right) = m\left(\left[\underline{x_{ij}^{l}}; \overline{x_{ij}^{l}}\right]\right) = \frac{\underline{x_{ij}^{l}} + \overline{x_{ij}^{l}}}{2} \tag{22}$$

Sendo *l* o índice do mapa de características e *i, j* as coordenadas de cada ponto. Desta forma, obtêm-se todos os *feature maps* com seus valores em ponto flutuante.

Até este ponto, ao final das camadas de convolução, foi realizado o controle de propagação do erro no percurso da rede com a utilização da aritmética intervalar. E mesmo alterando os dados para valores reais ainda pode ser percebido nos

resultados apresentados por Tortelli (2016) uma maior exatidão numérica e um controle na propagação do erro no decorrer da rede, comparando-se a dados obtidos puramente por cálculos na aritmética real.

Neste trabalho procura-se fazer uma comparação do erro associado na execução das convoluções da rede, com o uso da matemática intervalar e verificar a classificação obtida ao final da mesma.

Visando observar a qualidade de classificação e para nos apropriarmos mais dos valores presentes no intervalo de cada pixel, cada imagem intervalar no início do processo é fatiada ou repartida em n partes e gera n imagens convencionais por imagem intervalar abordada. Após o fatiamento, tem-se, de uma imagem *intervalar* original, o aumento para n imagens *reais* a serem inseridas na rede. Ou seja, com este fatiamento operacionaliza-se a representação da ideia da imagem intervalar e tem-se uma estratégia de aumento do conjunto de dados, uma vez que mais imagens seriam usadas para treino e classificação.

Pode-se observar a nova estrutura da rede na figura 42.

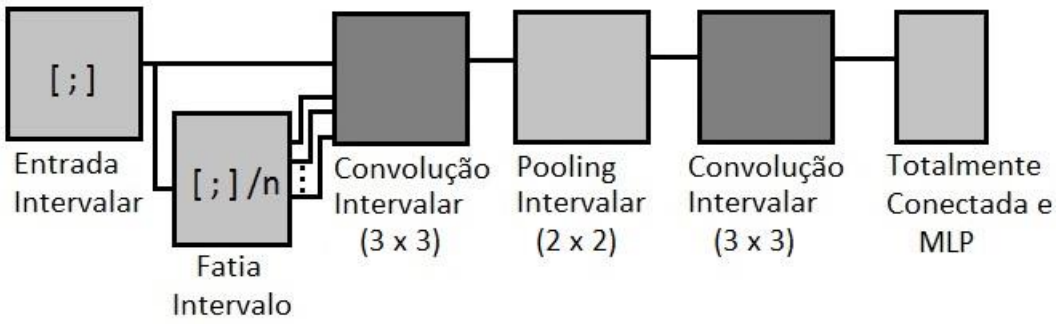


Figura 42 – Estrutura da rede CNN Intervalar deste trabalho
Fonte: Inspirada em Tortelli, 2016.

A ideia de fatiar o intervalo pode ser melhor compreendida observando-se a figura 43, onde tem-se um exemplo de pixel intervalar na imagem a ser fatiada e as respectivas imagens após este fatiamento. No exemplo, seriam 11 imagens resultantes de uma mesma imagem intervalar original.

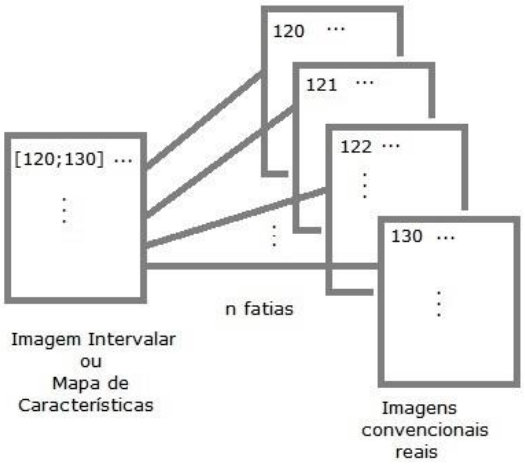


Figura 43 – Exemplo de fatiamento intervalar

Salienta-se que existem dois caminhos de execução, um para comparação e estudo do erro e outro para buscar a obtenção de uma melhor classificação final da rede, quando se perde o controle e acompanhamento do erro em prol de buscar um melhor resultado final de classificação.

A ferramenta selecionada para implementação deste trabalho, em continuidade a trabalhos desenvolvidos pelo grupo, foi a linguagem Python que possui biblioteca de compatibilidade com o tipo intervalar, IntPy. Esta é uma biblioteca gratuita e *opensource* que obteve melhores resultados em estudos comparativos com outras bibliotecas intervalares, como na exatidão numérica e tempo de processamento (BALBONI et al., 2014).

São utilizadas neste trabalho o Diâmetro (Equação 23), Erro Absoluto (Equação 24) e Desvio Padrão (Equação 25) como medidas de qualidade para a aritmética intervalar.

$$Diâmetro(X) = w(X) = \overline{x} - \underline{x} \tag{23}$$

$$Erro\ Absoluto = |x - m(X)| < \frac{w(X)}{2} \tag{24}$$

Sendo X a representação intervalar do valor real *x* e m(X) representa o ponto médio do intervalo X.

$$Dp = \sqrt{\frac{\sum_{i=1}^{n} (x_i - \bar{x})^2}{n-1}} \tag{25}$$

Onde $\bar{x}$ é a média aritmética da série, *n* é o tamanho da amostra, x_i é dado da série.

Outra métrica de qualidade que poderia ser considerada é a do Erro Relativo (Equação 26), porém as imagens utilizadas neste trabalho estão em *grayscale*[2] e podem conter pixels pretos (valor 0), tornando esta medida não recomendada.

$$Erro\ Relativo = \left|\frac{x-m(X)}{x}\right| \leq \frac{w(X)}{2min|x|}\ se\ 0\ \notin X \quad (26)$$

Para investigarmos a medida de qualidade dos mapas de características gerados a partir das camadas de convolução será utilizado o indicador de erro absoluto.

3.4 Resultados

Os testes foram realizados em duas etapas: para verificação da exatidão numérica, exibindo o controle do erro e para verificação dos acertos na classificação final, quando é inserida a ideia de fatiar a imagem intervalar para a geração de mais imagens convencionais e possível melhoria nos resultados nesta classificação.

Mantendo a conformidade com trabalho prévio do grupo a rede contém duas camadas de convolução 3x3 e uma camada de *pooling* 2x2 entre estas convoluções. Os filtros permaneceram os mesmos, fixos, na etapa de verificação de exatidão numérica, para uma avaliação adequada dos resultados decorrentes da execução em relação ao uso de intervalos e do mapeamento do erro propagado.

Ao final, estes resultados são inseridos em MLP disponibilizada pela biblioteca SciKit-Learn de algoritmos de aprendizagem, a qual utiliza os seguintes parâmetros para a realização da simulação:

- 200 iterações da Rede Neural por Convolução Intervalar, para o cálculo da acurácia por vizinhança e fatiamento.
- Taxa de aprendizado[3] constante.
- Método L-BFGS, algorítmo de otimização de Broyden Fletcher Goldfard Shanno, com memória limitada.
- Alpha $1x10^{-5}$. Na sciKit learn o alpha representa a taxa de aprendizado da rede.

Foi configurada uma rede de 5 camadas com 2 neurônios cada, na MLP.

Para a simulação de execução da rede foram utilizadas um total de 559 imagens no formato PNG, monocromáticas e com resolução espacial de 66×72,

[2] Ou Escalas de Cinza, imagens em tons de cinza.
[3] *Learning rate*

aplicando-se vizinhança de 8 e vizinhança de 4 para a geração de cada imagem de forma intervalar. Estas imagens estão agrupadas em três classes distintas: Carros(247), Mãos(160) e Prédios(152). O dataset de trabalho foi montado pelo grupo, com imagens de domínio público. A figura 44 exibe uma amostra destas imagens.

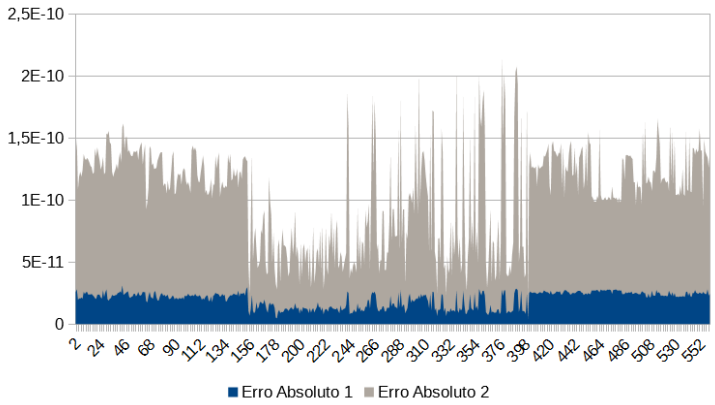
Figura 44 – Exemplo de imagens do dataset: (a) carro (b) mão e (c) prédio.

As simulações foram realizadas em computador com processador i5 de quinta geração, 3,4GHz, com 8Gb de memória RAM DDR3, sistema operacional Linux Ubuntu 15.04 e placa Intel HD Graphics 4000.

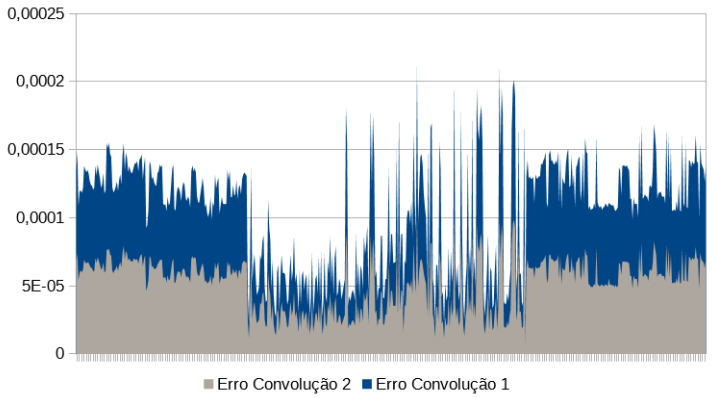
Para a representação de resultados de exatidão numérica da rede neural por convolução intervalar (figura 41) foram realizadas primeiramente apenas execuções com valores reais, tanto em imagens quanto em filtros, com a finalidade de obter dados sobre cada pixel dos features maps produzidos. Então foi simulada a implementação com valores intervalares, desde a imagem até os filtros das convoluções, com os mesmos parâmetros da execução real, para a obtenção de informação para o cálculo do erro absoluto que será um indicador de qualidade neste trabalho.

As figuras 45, 46 e 47 apresentam o cálculo do erro absoluto médio para cada imagem presente nas classes de prédios, mãos e carros. Cada figura contém dois gráficos, onde o primeiro representa a diferença entre o valor real e o intervalo e o segundo corresponde ao erro do intervalo gerado em cada camada de convolução.

Os resultados de exatidão numérica obtidos por Tortelli (2016) podem ser revistos na figura 45, lembrando que este utilizou imagens reais na entrada das convoluções.



(a) Erro do valor real em relação ao seu intervalo



(b) Erro do intervalo solução

Figura 45 – Gráfico da Média de Erro Absoluto de todas as classes
Fonte: Tortelli, 2016.

Os resultados de performance de exatidão numérica da rede, obtidos por Tortelli (2016), com a métrica de erro absoluto, podem ser observados na tabela 6.

Tabela 6 – Média do erro absoluto após as operações de convolução

	Erro Médio	Desvio Padrão
Convolução 1	$1{,}98 \times 10^{-11} < 0{,}0001038241$	$1{,}03 \times 10^{-12} < 2{,}41789481728 \times 10^{-6}$
Convolução 2	$1{,}05 \times 10^{-10} < 5{,}20 \times 10^{-5}$	$2{,}0136 \times 10^{-12} < 1{,}2212 \times 10^{-6}$

Fonte: Tortelli, 2016.

Os resultados de performance de classificação da rede obtidos por Tortelli (2016) podem ser observados na tabela 7.

Tabela 7 – Taxa de acertos da classificação das Redes Neurais Real e Intervalar

	Real	Intervalar
Classes Corretas	0,4995039683	0,4777777778
Desvio Padrão	0,1179531839	0,1303549708

Fonte: Tortelli, 2016.

A figura 46 começa a apresentar os resultados deste trabalho com imagens intervalares como entrada da rede por convolução intervalar, sendo neste com as imagens geradas por vizinhança de 8.

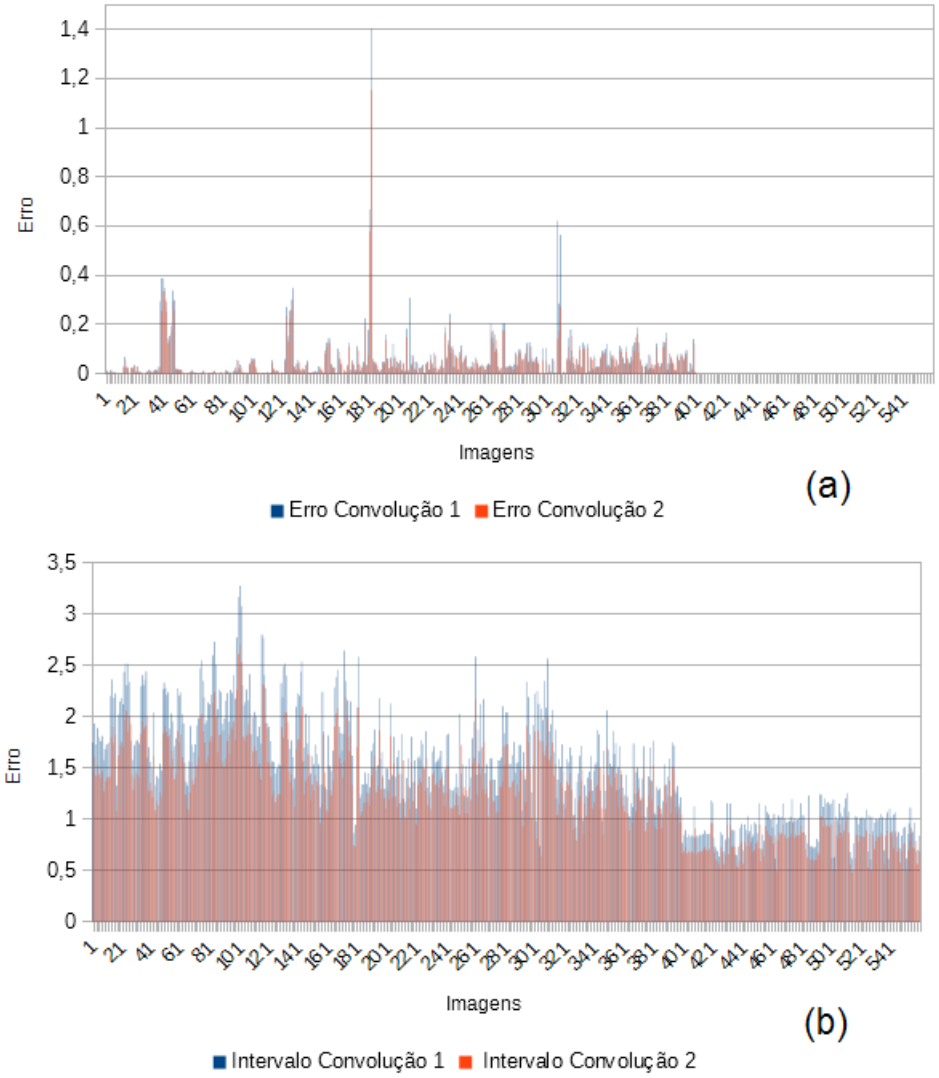


Figura 46 – Gráfico da média de erro absoluto de todas as classes, de imagens geradas por vizinhança de 8. (a) Erro do valor real em relação ao seu intervalo; (b) Erro do intervalo solução;

Lembrando que o $Erro\ Absoluto = |x - m(X)| < \frac{w(X)}{2}$, onde a primeira parte da inequação se refere ao valor real e o respectivo ponto médio do valor intervalar e a segunda parte da inequação se refere ao intervalo solução, ou seja, a metade do diâmetro do intervalo. No gráfico são apresentadas as médias destes valores, dos pixels das imagens.

A tabela 8 apresenta a média dos resultados de cálculo de erro absoluto obtidas na execução da rede com imagens intervalares e o desvio padrão percebido neste erro.

Tabela 8 – Média de erro absoluto após as operações de convolução em imagens intervalares geradas por vizinhança de 8.

	Erro Médio	Desvio Padrão
Convolução 1	$4{,}43 \times 10^{-2} < 1{,}5$	$9{,}47 \times 10^{-2} < 0{,}51946258$
Convolução 2	$3{,}52 \times 10^{-2} < 1{,}24726083$	$7{,}43 \times 10^{-2} < 0{,}42328541$

Observando-se a tabela 8 nota-se que os valores médios do erro absoluto diminuíram da convolução 1 para a convolução 2, bem como o diâmetro dos seus intervalos e respectivos desvios padrões das imagens geradas por vizinhança de 8.

A figura 47 apresenta o resultado deste trabalho, em relação ao controle de erro, com imagens intervalares geradas por vizinhança de 4 como entrada da rede por convolução intervalar.

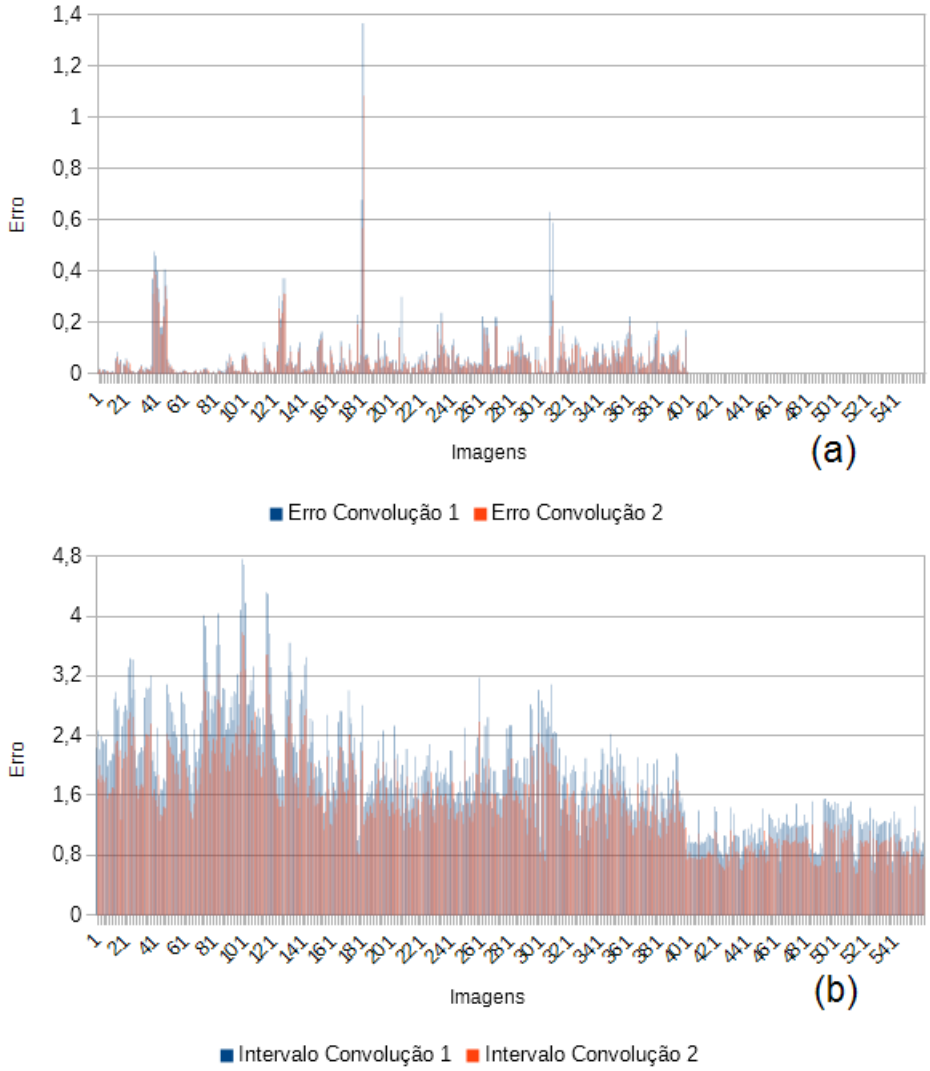


Figura 47 – Gráfico da média de erro absoluto de todas as classes, de imagens geradas por vizinhança de 4; (a) Erro do valor real em relação ao seu intervalo; (b) Erro do intervalo solução.

Na figura 47(a) pode-se perceber que o valor médio do erro absoluto de algumas imagens foi ínfimo, não sendo visíveis no gráfico. Nos dois gráficos também pode ser observada uma diminuição neste erro e também no diâmetro, na passagem da convolução 1 para a convolução 2.

A tabela 9 apresenta uma média dos resultados obtidos e o desvio padrão percebido.

Tabela 9 – Média de erro absoluto após as operações de convolução em imagens intervalares geradas por vizinhança de 4.

	Erro Médio	Desvio Padrão
Convolução 1	$5{,}10 \times 10^{-2} < 1{,}87$	0,09946876 < 0,72971580
Convolução 2	$3{,}98 \times 10^{-2} < 1{,}50523156$	0,07634868 < 0,57627798

Observa-se na tabela 9 a diminuição do erro médio absoluto de uma convolução para outra e estes respeitam o critério de qualidade referente ao tamanho do diâmetro do intervalo.

Verifica-se que os gráficos gerados após a convolução em imagens intervalares geradas por vizinhança de 4 ou de 8 são próximos em valores, mas não iguais, pois diferem na forma de obter a respectiva imagem intervalar.

A figura 48, juntamente com a tabela 10, apresenta os resultados de qualidade de classificação obtidos pela rede, não mais focando na precisão numérica ou controle de erro. Para isso, foi incluída a ideia de fatiamento da imagem intervalar de entrada, buscando aumentar a qualidade da classificação. Tem-se a imagem não fatiada ou fatiada por 1, a imagem fatiada por 3 e fatiada por 6 e suas respectivas médias de acerto nas classificações.

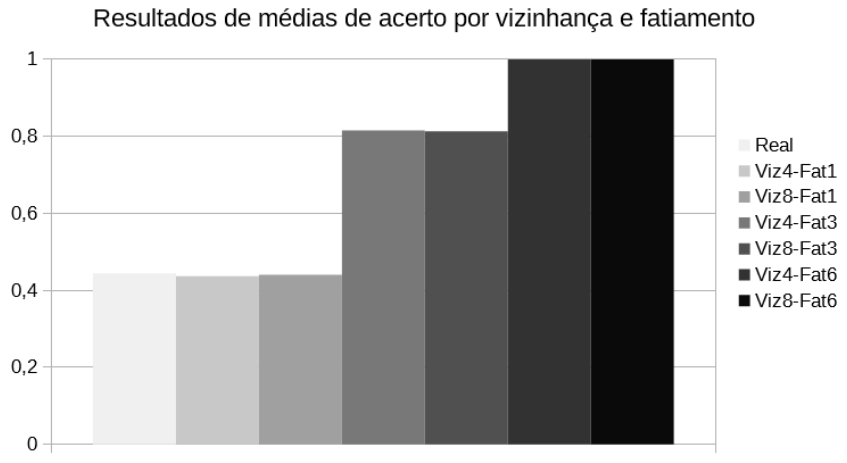


Figura 48 – Médias de classificação por vizinhança e fatiamento

A tabela 10 apresenta os valores das médias de classificação por situação e o referente desvio padrão.

Tabela10 – Média de Classificação

	Média de Acertos %	Desvio Padrão %
Real	44,44	2,33
Viz8-Fat1	44,05	2,16
Viz4-Fat1	43,65	2,44
Viz8-Fat3	81,35	1,87
Viz4-Fat3	81,55	1,73
Viz8-Fat6	100	0
Viz4-Fat6	100	0

Pode-se perceber considerando a figura 48 e a tabela 10 que as operações sobre as imagens geradas por vizinhança de 4 e de 8, respeitando o respectivo fatiamento, obtiveram valores próximos com relação a corretude na classificação. Porém considera-se que as operações de classificação sobre imagens geradas por vizinhança de 4 e fatiadas em 3 partes obtiveram um melhor resultado de classificação neste momento. O desempenho da generalização foi medido pelo conjunto de teste.

Sugere-se a ocorrência de *overfiting* quando ocorre o fatiamento por 6, indicando que a rede está sofrendo de sobreajuste e não predizendo valores corretamente. Isto pode se dar devido a se tratar com poucas imagens. Será utilizada a técnica de validação cruzada para confirmar esta suspeita.

Pode ser percebido que nas simulações que envolvem imagens geradas por vizinhança de 4 ou por vizinhança de 8 as percentagens de acertos são muito próximas.

Ainda é possível reconhecer uma pequena queda na média de acertos entre a rede neural com valores reais e a rede neural usando valores intervalares sem fatiamento, logo substancialmente corrigida no fatiamento por 3 das imagens intervalares, indicando que este fatiamento, possível de ser executado nestas, vem a fomentar uma maior acurácia em nosso modelo.

O desempenho em separado da generalização não foi medido de fato, foi realizada uma análise global do aprendizado, demonstrado pelos acertos e erros.

A validação cruzada é um método para avaliar classificadores e se baseia em dividir os dados em k grupos (ou *folds*) de tamanhos aproximadamente iguais, onde

um destes é deixado para teste e os demais para treinamento. Esse procedimento se repete k vezes, até que todas as partes tenham sido testadas (QUILICI-GONZALEZ; ZAMPIROLLI, 2014).

Aplicando o método de validação cruzada para validar os resultados previamente obtidos, utilizando 10 grupos na separação dos dados (10-*folds*) obtivemos a tabela 11.

Tabela 11 – Validação Cruzada

	Média de Acertos %	Desvio Padrão %
Viz8-Fat1	44,18	0,93
Viz4-Fat1		
Viz8-Fat3	81,58	1,18
Viz4-Fat3		
Viz8-Fat6	100	0
Viz4-Fat6		

Com esta execução confirma-se o *overfiting* quando a imagem é fatiada por 6. Nota-se também que os resultados de acurácia para imagens geradas por vizinhança de 4 e de 8 estão absolutamente iguais, indicando não haver preferência na utilização de um ou outro método para a obtenção de imagem intervalar no que concerne a classificação.

Conforme FAWZI (2016), *data augmentation*[4] é um processo de geração de amostras por transformação de dados de treinamento, com o objetivo de melhorar a precisão e robustez dos classificadores. Em nosso trabalho procuramos utilizar esta ideia e incluímos outro módulo em nossa estrutura de rede, alterando o modelo representado pela figura 42 para o novo modelo da figura 49.

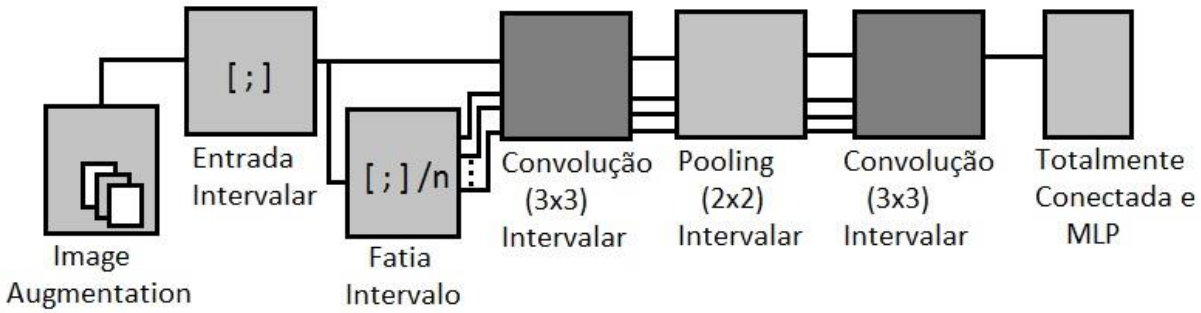


Figura 49 – Nova estrutura da rede CNN Intervalar

Os métodos utilizados foram de translação, blur, escala, flip horizontal e vertical. Na translação a imagem é movimentada sobre o eixo das abcissas. O blur é

[4] Aumento de dados

como se a imagem “borrasse”, fosse desfocada. A escala, neste trabalho, amplia partes da imagem. E o flip faz o giro da imagem, tanto no sentido horizontal como no vertical. Estas técnicas são utilizadas juntas, a fim de gerar novas imagens a partir das existentes.

A tabela 12 trará os resultados de acurácia obtidos com o uso das técnicas de aumento de dados, ou neste caso, *Image Augmentation*, aplicadas conforme figura 47, antes da geração de imagens intervalares. Também apresenta as configurações utilizadas nas técnicas abordadas Blur, Translação e Escala, visto as operações de flip terem ocorrido igualmente em todos os casos.

Tabela 12 – Utilizando técnicas de *Image Augmentation*

	Blur	Translação px (Eixo x)	Escala % (aumento)	Média de Acurácia %	Desvio Padrão %
Caso 1	-	-	-	43,57	1,10
Caso 2	1.0	25	100	44,18	1,11
Caso 3	2.0	25	40-60	44,22	0,73
Caso 4	3.0	-16	80-120	43,68	1,09
Caso 5	0.5	50	20	43,73	1,01
Caso 6	4.5	-70	90	44,10	1,38

Na tabela 12 o caso 1 usa filtro e imagens reais para as convoluções, para controle; o caso 2 ao caso 6 usa imagens e filtro intervalares, sem fatiamento, operando com imagens geradas por vizinhança de 4. Chegou a trabalhar com 1258 imagens tendo partido das 259 imagens iniciais, tendo aumentado os dados em mais de 4 vezes.

A média de acurácia entre os casos 2 e 6 ficou em 43,98% aparentemente apresentando pequena melhoria em relação a classificação de rede com imagens geradas por vizinhança de 4, sem fatiamento (43,65%).

A execução da rede, com a inclusão das técnicas de *Image Augmentation*, levou mais de 15 dias, o que demonstra como o processamento pode ser pesado para o tratamento da imagem, treinamento e teste de uma rede convolucional intervalar, bem como exigir um equipamento mais potente para a realização de todos os testes.

Na literatura científica foram encontradas algumas documentações consideradas descrições de estado da arte, porém nenhuma delas tratava com imagens intervalares e redes neurais artificias em conjunto. Dentre as encontradas que focavam em redes neurais convolutivas para o reconhecimento de padrões, foram selecionadas três devido a apresentação clara das respectivas acurácias, que seguem listadas a seguir.

Destaca-se o trabalho de Krizhevsky et al. (2012), que desenvolveram uma rede neural convolucional profunda para a classificação, com recursos da base de dados Imagenet, 1,2 milhões de imagens de alta resolução e 1000 diferentes classes. Utilizou 650.000 neurônios, 5 camadas de convolução, *max pooling,* entre outros recursos. Sua acurácia foi de 83% no melhor caso.

Outro estudo que pode ser salientado é o de Kukharenko e Konushin (2015), uma rede neural convolucional profunda que usa várias características da aparência das pessoas para classificação simultânea, a qual utilizou 13.233 imagens de faces com tamanho de 32 x 32, com 3.096 neurônios no nível de entrada da rede, 2 camadas de convolução com 64 filtros em cada e validação cruzada. Entre as características pesquisadas não alcançou acurácia inferior a 91,58%.

A última análise elencada foi realizada por Rocha (2015), cuja rede neural convolutiva para reconhecimento de objetos foi testada na base de dados CIFAR-10 com 60.000 imagens de tamanho 32 x 32, divididas em 10 classes. Utilizou-se a linguagem de programação Java, com biblioteca em Python. Trabalhava com 1 camada de convolução 3x3, *max pooling* e 100 neurônios na camada de saída. A acurácia de seu pior modelo foi de 74,19% e de seu melhor, 87,14%.

Na atual proposta de rede neural convolucional intervalar tem-se 10 neurônios e 2 camadas de convolução 3x3. Trabalhou-se com 259 até 1.258 imagens, de 66 x 72, *max pooling* e Python. O mínimo de acurácia atingido foi de 43,57% e o máximo, 81,55%.

Com isto, confirma-se a probabilidade de a quantidade de neurônios ser proporcional à acurácia da rede em alguns casos, ou seja, enquanto esta proposta possui o menor número de neurônios (10), também é a que possui a melhor acurácia mais baixa (81,55%), porém a rede proposta por Kukharenko e Konushin (2015) possui a pior acurácia (91,58%) mais alta que a melhor acurácia (83%) de Krizhevsky et al. (2012), que possuem milhares de vezes mais neurônios, o que resulta na conclusão de que o número de neurônios não é o único aspecto

determinante de uma melhor acurácia. Porém, apesar destas análises, é importante ressaltar que os valores das acurácias entre estes estudos não se discreparam de forma relevante.

4 CONCLUSÃO

O uso da matemática intervalar garante o controle do erro no processamento efetuado nas convoluções da rede neural intervalar. Verifica-se que com o fatiamento das imagens intervalares na entrada da rede obteve-se uma melhor qualidade na classificação das imagens. Recurso aplicável somente às imagens de representação intervalar.

Pode-se observar que o erro médio diminuiu de uma convolução para outra, assim como o desvio padrão, tanto com imagens geradas por vizinhança de 8 como por vizinhança de 4.

Percebe-se também que a média de acertos com vizinhança de 4 e fatiamento por 3 foi melhor que no equivalente de vizinhança de 8, caso não se observe os resultados da validação cruzada, que sugerem igual desempenho gerando-se as imagens por vizinhança de quatro ou por de oito. Estes confirmam o *overfiting* no fatiamento por 6. De todas as formas, o uso de fatiamento proporcionou um salto evidente na acurácia das classificações, algo que nem o uso de técnicas de *image augmentation* conseguiu superar.

Como trabalho futuro pode ser investigado o comportamento da rede com fatiamento por 5 e observar seu nível de acurácia, buscando aproximar-se do ponto de maior acurácia sem a ocorrência de *overfiting*.

É possível também alterar a base de dados e verificar a resposta da rede, pois as imagens intervalares dependem das vizinhanças de pixels para serem geradas e consequentemente são dependentes fortemente das imagens originais.

Ainda como seguimento deste trabalho pode-se ampliar os testes realizados com o aumento nos dados[5], neste caso imagens, para buscar ainda uma maior qualidade de classificação da rede, abrangendo os demais casos de fatiamento.

Sugere-se como implementações futuras usar novas métricas intervalares, como distância euclidiana ou erro quadrado médio (SANTOS, 2016) e verificar se há impacto sobre resultados obtidos, bem como averiguar mudanças no tempo de processamento.

Também pode-se usar filtros diferentes por fatia, n filtros arrolados para destacar características específicas nas imagens analisadas.

[5] *Image Augmentation de Data Augmentation*

Ainda considerar em utilizar o valor médio dos diâmetros dos intervalos das imagens para decidir se ocorrerá fatiamento ou não, ou mesmo a quantidade de fatias que a respectiva imagem poderá fornecer.

Atualmente não dispomos de MLP intervalar, mas futuramente ao se ter uma rede neural intervalar implementada, poderá se realizar todo o processo, desde a geração de imagens intervalares até a classificação final da rede, utilizando apenas valores intervalares, garantindo assim sua exatidão de início a fim (ESCARCINA et al., 2004).

Encontrou-se dificuldade em relacionar duas áreas tão diferentes, como matemática intervalar e redes neurais convolucionais, pois não foram encontrados trabalhos que as abordem em conjunto, com exceção deste e da versão de Tortelli.

REFERÊNCIAS

ACIÓLY, B.M.; BEDREGAL, B.R.C. A quasi-metric topology compatible with inclusion monotonicity on interval space. **Reliable Computing**, 3, p. 305–313, 1997.

ALEFELD, G; HERZBERGER, J. **Introduction to Interval Computations.**New York: Academic Press, 1983. 332p.

BALBONI, M.; TORTELLI, L.; LORINI, M.; FURLA, V.; FINGER, A.; LORETO, A. Critérios para Análise e Escolha de Ambientes Intervalares. **Revista Jr de Iniciação Científica em Ciências Exatas e Engenharia, Rio Grande**, v.7, n.1, 2014.

BARRETO, Jorge M. Introdução as redes neurais artificiais. **V Escola Regional de Informática. Sociedade Brasileira de Computação,** Regional Sul, Santa Maria, Florianópolis, Maringá, p. 5-10, 2002.

BEDREGAL, B. R. C.; DORIA NETO, A.D.; BEDREGAL, R.C.; LYRA, A. O Construtor Intervalar; Morfologia Matemática e Imagens Digitais Intervalares. In: XXVI Congresso Nacional de Matemática Aplicada e Computacional, 2003, São José do Rio Preto. **Anais**... 2003, p.610.

BRAGA, A. P.; CARVALHO, A. P. L. F.; LUDERMIR, T. B. **Redes Neurais Artificiais: Teoria e Aplicações.** RJ: Editora LTC, 2000. 262p.

CLAUDIO, Dalcídio Moraes. MARINS, Jussara M. **Cálculo Numérico computacional.** São Paulo: Ahals, 1994. 464p.

CRUZ, M.M.C.; SANTIAGO, R.H.N.; BEDREGAL, E.R.C.; DORIA NETO, A.D.D. Um modelo intervalar aplicado à Morfologia matemática. In: XXXIII Congresso Nacional de Matemática Aplicada e Computacional, 2010, Águas de Lindóia, SP. **Anais**... SBMAC 2010, p.123-129.

CRUZ, M.M.C.; SANTIAGO, R.H.N. Morfologia Intervalar X Morfologia Fuzzy: Uma AnáliseComparativa In: I Congresso Nacional de Matemática Aplicada e Computacional da Região Sudeste, 2011, Águas de Lindóia, SP. **Anais**... SBMAC 2011, p.585-588.

CRUZ, Marcia Maria de Castro. **Uma Fundamentação Intervalar Aplicada àMorfologia Matemática.** 128 f. Tese (Doutorado em Ciências), Programa de Pós-Graduação em Engenharia Elétrica e Computação. Universidade Federal do Rio Grande do Norte, Natal, 2008.

DE ALBUQUERQUE, Márcio Pontes; DE ALBUQUERQUE, Marcelo Pontes. Processamento de imagens: métodos e análises. **Centro Brasileiro de Pesquisas Físicas MCT**, 2000.

DE QUEIROZ, José Eustáquio Rangel; GOMES, Herman Martins. Introdução ao Processamento Digital de Imagens. 2006. **RITA, Revista de Informática Teórica e Aplicada**. UFGRS. v. 13, n. 2, p. 11-42.

ESCARCINA, R.E.; BEDREGAL, B.R.; LYRA, A. Redes neurais intervalares de uma camada. In: XXVII Congresso Nacional de Matemática Aplicada e Computacional, 2004, Porto Alegre, RS. **Anais...** SBMAC 2004, p.491.

ESQUEF, Israel Andrade. **Técnicas de entropia em processamento de imagens.** 2002. 155p. Dissertação (Mestrado em Instrumentação Científica), CBPF - Centro Brasileiro de Pesquisas Físicas, Rio de Janeiro.

ESQUEF, Israel Andrade; ALBUQUERQUE, Márcio Portes de; ALBUQUERQUE, Marcelo Portes de. Processamento Digital de Imagens. **CBPF - Centro Brasileiro de Pesquisas Físicas**, 18/02/2003.

FAUZI, Alhussein; SAMULOWITZ, Horst; TURAGA, Deepak; FROSSARD, Pascal. Adaptive data augmentation for image classification. In: **Image Processing (ICIP), 2016 IEEE International Conference on.** IEEE, 2016. p. 3688 – 3692.

FORESTI, Renan Luís.**Sistema de visão robótica para reconhecimento de contornos de componentes na aplicação de processos industriais.**2006. 64p. Dissertação (Mestrado em Engenharia). Escola de Engenharia da Universidade Federal do Rio Grande do Sul, Porto Alegre.

GONZALEZ, Rafael C.; WOODS, Richard E. **Digital Image Processing.** ThirdEdition. Pearson – Prentice Hall.2007, 976p.

GONZALEZ, Rafael C.; WOODS, Richard E. **Processamento de Imagens Digitais.**São Paulo: Blucher. 2010, 509p.

HAYKIN, S. **Redes Neurais: princípios e prática.** 2.ed. Porto Alegre: Bookman, 2001. 900p.

JURASZEK, Guilherme de Freitas. **Reconhecimento de produtos por imagem utilizando palavras visuais e redes neurais convolucionais**. 2014. 151f. Dissertação (Mestrado em Computação Aplicada) Programa de Pós-Graduação em Computação Aplicada. Universidade do Estado de Santa Catarina. Joinville.

KEARFOTT, R. Baker; KREINOVICH, Vladik. Applications of interval computations: An introduction. In: **Applications of Interval Computations**. Springer US, 1996. p. 1-22.

KRIZHEVSKI, Alex; SUTSKEVER, Ilya; HINTON, Geoffrey E. Imagenet classification with deep convolutional neural networks. In: **Advances in neural information processing systems.** 2012. p. 1097 – 1105.

KUKHARENKO, A.; KONUSHIN, A. Simultaneous classification of several features of a person's apperance using a deep convolutional neural network. In: **Pattern recognition and image analysis**. 2015. vol.25 (3), p. 461 – 465.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. **Nature**, New York, v.521, n.14539, p. 436 – 444, 2015.

LECUN, Y.; KAVUKCUOGLU, K.; FARABET, C. Convolutional networks and applications in vision. In: **Circuits and Systems (ISCAS)**. IEEE International Symposium, p. 253-256. 2010.

LYRA, A. **Uma fundamentação matemática para o processamento de imagens digitais intervalares.** 2003. 183f. Tese (Doutorado em Ciências) – Curso de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Rio Grande do Norte, Natal.

LYRA, A.; BEDREGAL, B.R.C.; DORIA NETO, A.D.; BEDREGAL, B.R.C. The Interval Digital Images Processing. **WSEAS TransactionsonCircuitsand Systems**, v. 3, n. 2, p. 229-233, 2004.

LYRA, A.; RIBEIRO, A.G.S.; BEDREGAL, B.R.C.; DORIA NETO, A.D. Um aplicativo computacional sob as Imagens Digitais Intervalares. In: XXVI Congresso Nacional de Matemática Aplicada e Computacional, 2003, São José do Rio Preto. **Anais**... 2003, p.93.

MARQUES FILHO, Ogê; NETO, Hugo Vieira. **Processamento digital de imagens**. Rio de Janeiro: Brasport. 1999. 307p.

MERTES, Jacqueline Gomes. **Implementação em FPGA de um sistema para processamento de imagens digitais para aplicações diversificadas.** 2012. 134p. Dissertação (Mestrado em Ciências da Computação) Sistemas de Computação. UNESP, Universidade Estadual Paulista, São José do Rio Preto.

MESQUITA, Marcos Paulo de. **Matemática intervalar: Princípios e a Ferramenta C-XSC.** 2002. 41F. Trabalho Individual (Bacharel em Ciências da Computação), Departamento de Ciências da Computação. Universidade Federal de Lavras, Minas Gerais.

MOORE, R. E. **Interval Analysis**. Englewood Cliffs: Prentice Hall, 1966.

NASCIMENTO, Tiago Cordeiro de Melo. **Segmentação de imagens utilizando elementos de morfologia matemática.** 2013. 55p. Trabalho de Conclusão de Curso (Bacharelado em Engenharia da Computação). Centro de Informática, Universidade Federal de Pernambuco, Recife.

NEVES, Samuel Clayton Maciel; PELAES, Evaldo Gonçalves. Estudo e implementação de técnicas de segmentação de imagens. **Revista Virtual de Iniciação Acadêmica da UFPA,** vol1, No 2, p.1-11, julho2001.

OLIVEIRA, Paulo Werlang; DIVÉRIO, Tiarajú Asmuz; CLÁUDIO, Dalcídio Moraes. **Fundamentos da matemática intervalar**, 1ª Ed. Porto Alegre: Sagra Luzzato. 1997. 93p.

OTUYAMA, Júlio Massayuki. **Rede Neural por Convolução para Reconstrução Estéreo.** 2000. 76f. Dissertação (Mestrado em Ciências da Computação) Centro Tecnológico, Universidade de Santa Catarina, Florianópolis.

QUILICI-GONZALEZ, José Artur; ZAMPIROLLI, Francisco de Assis. **Sistemas Inteligentes e Mineração de dados.** Santo André: Triunfal Gráfica e Editora, 2014. 148p.

RIBEIRO, Alan Gustavo Santana. **Um aplicativo computacional para o tratamento de imagens intervalares.** 2003. 51f. Monografia (Bacharelado em Sistemas de Informação) – Curso de Bacharelado em Sistemas de Informação, Universidade Potiguar, Natal.

RIBEIRO, Alan Gustavo Santana; LYRA, Aarão. PDI² – Processamento digital de imagens intervalares: Criação e tratamento deimagens intervalares. In: XXVII Congresso Nacional de Matemática Aplicada e Computacional, 2004, Porto Alegre, RS. **Anais SBMAC**... 2004, p.497.

RIBEIRO, Amado Emanoel. **Métodos de análise de microcospia do clínquer usando morfologia matemática.** 2003. 48p. Trabalho de Conclusão de Curso (Bacharelado em Ciências da Computação) Curso de Ciências da Computação da Universidade Presidente Antônio Carlos, Barbacena, MG.

ROCHA, Rafael Henrique Santos. **Reconhecimento de objetos por redes neurais convolutivas.** 2015. 49f. Dissertação (Mestrado) – Programa de pós-graduação em otimização e raciocínio automático, departamento de informática do centro técnico científico da PUC-Rio. Rio de Janeiro. 2015.

RUSSEL, S., NORVIG, P.; **Inteligência Artificial,** 3ª Ed. Campus, Elsevier, Rio de Janeiro, p. 990 ,2013.

SANTANA, Fagner Lemos de. **Generalizações do conceito de Distância, i-Distâncias, Distâncias Intervalares e Topologia**. 2012. 93p. Tese (Doutorado em Ciências). Programa de Pós-Graduação em Sistemas e Computação da Universidade Federal do Rio Grande do Norte, Natal, RN.

SANTOS, Daniel Rodrigues dos. **Extração semi-automática de edificações com análise do modelo numérico de elevações**. 2002. 166p. Dissertação (Mestrado em Ciências Cartográficas) Faculdade de Ciências e Tecnologia – UNESP, Presidente Prudente.

SANTOS, José Medeiros dos. **Em direção a uma representação para equações algébricas: uma lógica equacional local**. 2001. 161p. Dissertação (Mestrado em Sistemas e Computação), Centro de Ciências Exatas e da Terra. Departamento de Informática e Matemática Aplicada. Universidade Federal do Rio Grande do Norte, Natal, RN.

SANTOS, Vinícius Rodrigues dos. **Int-DWTs Library – Algebraic Simplifications Increasing Performance and Accuracy of Discrete Wavelet Transforms**. 59f. Trabalho de conclusão de curso (Graduação). Centro de desenvolvimento tecnológico da Universidade Federal de Pelotas, Pelotas, 2016.

SOUZA, F. A. A.; **Análise e desempenho da rede neural artificial do tipo multilayerperceptron na era multicore.** 2012. 49f. Dissertação (Mestrado em Ciências) Programa de Pós-Graduação em Engenharia Elétrica e de Computação. Universidade Federal do Rio Grande do Norte. Natal.

TAKAHASHI A., BEDREGAL, B. R. C.; LYRA, A. Uma versão intervalar do método de segmentação de imagens utilizando o k-means. In: **TEMA Trends in Applied Computational Mathematics** vol.6, n.2, 2005, p.315–324.

TAKAHASHI A.; BEDREGAL, B.R.C.; LYRA, A. Uma versão intervalar do método de segmentação de imagens utilizando o k-means. In: XXVII Congresso Nacional de Matemática Aplicada e Computacional, 2004, Porto Alegre, RS. **Anais SBMAC**... 2004, p.499.

TORTELLI, Lucas Mendes. **Extensão intervalar aplicado em Redes neurais por convolucão.** 2016. Monografia (Bacharelado em Ciências da Computação) – Centro de Desenvolvimento Tecnológico da Universidade Federal de Pelotas.

TRINDADE, Roque Mendes Prado. **Uma fundamentação matemática para proocessamento digital de sinais intervalares.** 2009. 197p. Tese (Doutorado em Ciências), Programa de Pós-Graduação em Engenharia Elétrica e Computação, Universidade Federal do Rio Grande do Norte, Natal, RN.

VARGAS, Rogério Rodrigues de. **Matemática intervalar aplicada ao problema do fluxo de potência com incertezas em redes de energia elétrica.** 2006. 39p. Trabalho Individual, Escola de Informática, Universidade Católica de Pelotas. Rio Grande do Sul.

VON WANGENHEIM, Aldo. **Introdução à Visão Computacional** - Encontrando a Linha Divisória: Detecção de Bordas. 2008. Apostila Disponível em: <https://www.inf.ufsc.br/~visao/bordas.html> Acesso em: 08/10/2015.