UNIVERSIDADE FEDERAL DA BAHIA
ESCOLA POLITÉCNICA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

APLICAÇÕES DE CNN ANALÓGICA EM TECNOLOGIA CMOS EM OPERAÇÕES DE FILTRAGEM PARA PROCESSAMENTO VISUAL

Fabian Souza de Andrade
Orientadora: Profª. Drª. Ana Isabela Araújo Cunha

Salvador
2020

FABIAN SOUZA DE ANDRADE

APLICAÇÕES DE CNN ANALÓGICA EM TECNOLOGIA CMOS EM OPERAÇÕES DE FILTRAGEM PARA PROCESSAMENTO VISUAL

Tese apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal da Bahia como parte dos requisitos necessários para obtenção do grau de Doutor em Engenharia Elétrica.

Orientadora:

Profª. Drª. Ana Isabela Araújo Cunha

Salvador, Bahia, Brasil

2020

Ficha catalográfica elaborada pelo Sistema Universitário de Bibliotecas (SIBI/UFBA), com os dados fornecidos pelo(a) autor(a).

Andrade, Fabian Souza de
Aplicações de CNN Analógica em Tecnologia CMOS em Operações de Filtragem para Processamento Visual / Fabian Souza de Andrade. -- Salvador, 2020.
131 f. : il

Orientadora: Profª. Drª. Ana Isabela Araújo Cunha.
Tese (Doutorado - Engenharia Elétrica) -- Universidade Federal da Bahia, Escola Politécnica, 2020.

1. Visão Artificial. 2. Circuitos Elétricos. 3. Redes neurais (Computação). 4. Filtragem de Imagens. I. Cunha, Profª. Drª. Ana Isabela Araújo. II. Título.

Fabian Souza de Andrade

“APLICAÇÕES DE CNN ANALÓGICA EM TECNOLOGIA CMOS EM OPERAÇÕES DE FILTRAGEM PARA PROCESSAMENTO VISUAL”

Tese apresentada à Universidade Federal da Bahia, como parte das exigências do Programa de Pós-Graduação em Engenharia Elétrica, para a obtenção do título de Doutor.

APROVADA em: 22 de Dezembro de 2020.

BANCA EXAMINADORA

Profª. Drª. Ana Isabela Araújo Cunha
Orientador/UFBA

Prof. Dr. Delmar Broglio Carvalho
UEFS

Prof. Dr. Edson Pinto Santana
UFBA

Prof. Dr. Eduardo Furtado de Simas Filho
UFBA

Prof. Dr. Eduardo Telmo Fonseca Santos
IFBA

Prof. Dr. Jés de Jesus Fiais Cerqueira
UFBA

DEDICATÓRIA

Aos meus pais, às minhas irmãs e à minha família.

AGRADECIMENTOS

Gostaria de agradecer imensamente à Profª. Drª. Ana Isabela Cunha, pela valorosa orientação exercida de forma tão participativa e atenciosa, me inspirando a desenvolver o trabalho buscando sempre o nível mais elevado. Ao Prof. Dr. Edson Santana, que além de ser o responsável pelo trabalho que serviu como ponto de partida do tema tratado nesta tese, esteve sempre disposto a contribuir de forma significativa para a sua evolução. Aos colegas do Laboratório de Concepção de Circuitos Integrados (LCCI) da UFBA, cujo ambiente de harmonioso convívio e intensa colaboração favoreceu amplamente a elaboração desta tese. Ao Prof. Dr. Eduardo Simas, pela participação relevante dado o caráter multidisciplinar desta obra. Aos demais professores, colegas e funcionários do Programa de Pós-Graduação em Engenharia Elétrica e do Departamento de Engenharia Elétrica da UFBA, pelo suporte exercido direta ou indiretamente.

Transmito igualmente os meus agradecimentos à minha família, cujo apoio irrestrito durante todo o período de desenvolvimento deste trabalho me permitiu chegar ao seu término através de muito empenho e repleto de motivação.

Ademais, sou grato à Fundação de Amparo à Pesquisa do Estado da Bahia (FAPESB) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), pelo aporte financeiro concedido para a elaboração desta Tese.

RESUMO

Este trabalho é uma contribuição na busca pela realização em tecnologia CMOS de um circuito analógico, pertencente à classe biomórfica das redes neuronais celulares (*cellular neural networks - CNN*), para o processamento de imagens. Visando a reprodução de operações mais complexas do que as usuais, uma arquitetura compacta de célula foi empregada na implementação de uma rede com duas camadas e acoplamento mútuo, a partir de uma extensão da CNN simples. A fim de se obterem os parâmetros que configuram a rede para executar as funções desejadas, uma metodologia de treinamento foi desenvolvida, inspirando-se em algumas técnicas existentes voltadas para esta categoria de rede neuronal e incluindo modificações para aprimorar tanto o seu desempenho de forma geral, quanto para adaptar o processo a algumas características relevantes dos diversos tipos de operações sobre imagens considerados. Como forma de verificação, o circuito é aplicado em simulações envolvendo funções bipolares e filtragem de imagens, apresentando resultados similares aos gerados por um modelo ideal.

Palavras-chave: Rede neuronal, CNN, CMA, Algoritmo Genético, Filtragem de Imagens.

ABSTRACT

This work is a contribution to the realization in CMOS technology of an analog circuit, belonging to the biomorphic class of cellular neural networks, for image processing. Aiming at the reproduction of more complex operations than usual, a compact cell architecture was employed in the implementation of a network featuring two layers and mutual coupling, originated from an extension of the traditional CNN. To obtain the parameters that configure the network to perform the desired functions, a training methodology was developed, taking inspiration from some existing techniques addressed to this category of neuronal network and including modifications either to improve its performance in general or to adapt the process to some relevant characteristics of the various types of image operations considered. For verification purposes, the circuit is applied in simulations involving bipolar functions and image filtering, exhibiting similar results to the ones generated from an ideal model.

Keywords: Neural network, CNN, CMA, Genetic Algorithm, Image Filtering.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

CMOS	*Complementary Metal–Oxide–Semiconductor*
RNA	Redes Neuronais Artificiais
CNN	*Cellular Neural Network*
FSR	*Full Signal Range*
FFT	*Fast Fourier Transform*
2L-CNN	*Two-Layers Cellular Neural Network*
1L-CNN	*One-Layer Cellular Neural Network*
2LFF-CNN	*Two-Layers Feedforward Cellular Neural Network*
2LFB-CNN	*Two-Layers Feedback Cellular Neural Network*
CMA	*Center of Mass Algorithm*
GA	*Genetic Algorithm*
DFT-2D	*Two Dimensional Discrete Fourier Transform*
IDFT-2D	*Two Dimensional Inverse Discrete Fourier Transform*
DC	Corrente Contínua
RAM	*Random Access Memory*
LCCI	Laboratório de Concepção de Circuitos Integrados
UFBA	Universidade Federal da Bahia

LISTA DE SÍMBOLOS

M	Número de linhas da CNN/imagem
N	Número de colunas da CNN/imagem
$Ce(i,j)$	Célula localizada nas coordenadas (i,j)
S_R	Esfera de influência da CNN
R	Raio de S_r
$x_{i,j}$	Estado de $Ce(i,j)$
$y_{i,j}$	Saída de $Ce(i,j)$
$u_{i,j}$	Entrada de $Ce(i,j)$
$z_{i,j}$	Limiar de $Ce(i,j)$
A	Operador sináptico de realimentação
B	Operador sináptico de entrada
t	Tempo
$a_{m,n}$	Coeficiente da posição (m,n) em A
$b_{m,n}$	Coeficiente da posição (m,n) em B
$gp(x)$	Função de grampeamento da célula do tipo FSR
A_{FSR}	Operador sináptico de realimentação da célula do tipo FSR
$Ce(i,jm)$	Célula localizada nas coordenadas (i,j) da camada m
C	Operador sináptico de acoplamento
e_{ij}	Erro de $Ce(i,j)$
d_{ij}	Resposta desejada de $Ce(i,j)$
k	Índice da iteração
η	Taxa de aprendizado
$b_{m,n}$	Coeficiente da posição (m,n) em B
$\Delta a_{m,n}$	Variação de $a_{m,n}$
$\Delta b_{m,n}$	Variação de $b_{m,n}$
$\Delta z_{m,n}$	Variação de $z_{m,n}$
A_S	Componente simétrica de A
A_A	Componente anti-simétrica de A
A_C	Componente complementar da decomposição de A
ΔA_S	Variação de A_S

ΔA_A	Variação de A_A
$m_{i,j}$	Massa de $Ce(i,j)$
r_l	Centro de massa em relação ao eixo l
M_t	Massa total da rede
$d(l)_{ij}$	Distância entre $Ce(i,j)$ e o eixo l
$ct(p)$	Custo associado ao indivíduo p
ft	Aptidão
$maxdiff$	Maior diferença possível entre d e y
np	Número de indivíduos
Ps_{in}	Probabilidade de seleção do indivíduo in
$f(x,y)$	Valor do pixel na coordenada (x,y) antes da filtragem
$g(x,y)$	Valor do pixel na coordenada (x,y) após a filtragem
$w(m,n)$	Coeficiente da posição (m,n) do filtro espacial
$\tilde{F}(\omega,\psi)$	Transformada discreta de Fourier de $f(x,y)$
$\tilde{H}(\omega,\psi)$	Função de transferência do filtro espacial
$\tilde{H}_{PB}(\omega,\psi)$	Função de transferência do filtro espacial passa-baixas
Do	Frequência de corte do filtro
$D(\omega,\psi)$	Distância de um ponto (ω,ψ) para o centro do filtro
$\tilde{H}_{PA}(\omega,\psi)$	Função de transferência do filtro passa-altas
P	Dimensão horizontal da imagem ampliada
Q	Dimensão vertical da imagem ampliada
$\tilde{\mathrm{X}}_\infty$	Transformada de discreta de Fourier dos estados $x_{i,j}$ no regime permanente
i_{outA}	Corrente de saída do multiplicador
i_{in}	Corrente de entrada do multiplicador
I_B	Corrente de polarização
V_{DS1}	Tensão entre dreno e fonte do transistor
V_{IDC}	Deslocamento de nível total
v_{ish}	Sinais de tensão de estado e de entrada na célula
v_o	Saída de tensão da célula
Δt	Passo temporal
$ep_{l,ij}$	Erro propagado da célula $Ce(i,j)$ da camada l
ρ	Coeficiente da CNN

$\hat{\rho}$	Coeficiente da CNN normalizado
F_N	Fator de normalização
F_N	Fator de normalização
f_p	Função densidade de probabilidade
g_a	Geração atual do GA
n_g	Número máximo de gerações do GA
θ	Parâmetro do operador de mutação
$\tilde{y}_{i,j}$	Saída da célula $Ce(i,j)$
$y_{t_{i,j}}$	Saída da célula $Ce(i,j)$ no tempo t
$y_{\infty_{i,j}}$	Saída da célula $Ce(i,j)$ em regime permanente
$\tilde{X}_{l_\infty}$	Transformada de discreta de Fourier dos estados $x_{i,j}$ da camada l no regime permanente
k_a	Fator multiplicativo para compensação de assimetria
I_Z	Corrente para compensação de offset
Im_E	Imagem de entrada
Im_S	Imagem de saída
E_{SM}	Erro relativo entre a rede simulada e o modelo ideal
E_{SF}	Erro relativo entre a rede simulada e a filtragem teórica
E_{MF}	Erro relativo entre o modelo ideal e a filtragem teórica

SUMÁRIO

1 INTRODUÇÃO

A computação analógica a partir de sistemas eletrônicos é um conceito existente há muitas décadas, tendo seus primeiros passos ocorridos na década de 1940 com implementações a válvulas, sendo de grande importância para variadas aplicações. Contudo, havia uma apreciável dificuldade atrelada à sua operação, que exigia profissionais especialmente treinados. O surgimento dos transistores, seguido pelo desenvolvimento de circuitos integrados, permitiu o advento dos sistemas digitais, que traziam múltiplas vantagens, dentre as quais podem-se destacar a programabilidade intuitiva e a alta precisão. Tais sistemas representaram assim uma alternativa mais viável para a computação, e sua evolução culminou no declínio substancial da abordagem analógica com o passar do tempo. Mesmo as tentativas de se utilizar sistemas híbridos não conseguiram dar uma sobrevida longa a essa linha.

Todavia, nas últimas décadas, a demanda pelo desenvolvimento de ambientes mais conectados e automatizados tem motivado a concepção de dispositivos eletrônicos e redes de sensores mais rápidos, compactos e eficientes. Além disso, o crescente interesse em aplicações biomédicas, como dispositivos implantáveis, reforça a busca por arquiteturas que funcionem da forma mais próxima possível dos sistemas biológicos. Nesse contexto, despontou-se uma tendência para o retorno da computação analógica (CHUA e ROSKA, 2002), tirando proveito principalmente dos grandes avanços na tecnologia *complementary metal–oxide–semiconductor* (CMOS).

Um campo que se destaca nessa situação é o processamento visual, cada vez mais relevante em um mundo que recorre aos dados capturados por câmeras densamente presentes para fins de reconhecimento, inspeção ou vigilância, muitas vezes de forma automatizada. Exemplos bem consolidados neste contexto são a identificação de impressões digitais (KAMEI e MIZOGUCH, 1995) e de caracteres (TAVSANOGLU e SAATCI, 2000), bem como diagnóstico médico (NIU, SHEN, *et al.*, 2010), (PLEBE e GALLO, 2001), onde se utiliza tipicamente operações como o ajuste dos níveis de contraste, a remoção de partes da imagem e a aplicação de filtros, cujos efeitos podem variar desde a suavização até o realce de bordas (GONZALEZ e WOODS, 2007).

Nesta classe de aplicações, destaca-se a forte presença de redes neuronais artificiais (RNA), modelos biomórficos inspirados pela organização das células nervosas, em geral implementados por circuitos digitais, que conseguem exercer satisfatoriamente funções bem complexas, contando com a vantagem provida pelo alto paralelismo

intrínseco à sua estrutura distribuída. A RNA convolucional é um exemplo específico bem apto para essa tarefa, já que seu funcionamento se assemelha bastante do ponto de vista matemático ao processo de filtragem espacial (CUN, MATAN, *et al.*, 1990). Soma-se a esses aspectos a conveniente possibilidade de utilizar a própria capacidade de inteligência computacional de uma RNA para a realização, de forma integrada, de classificações ou tomadas de decisão, contribuindo para a autonomia dos sistemas.

Adicionalmente, a origem biológica dessa estratégia pode ser bem favorável para a criação de sistemas conectados diretamente com estruturas celulares, como próteses e dispositivos implantáveis, apresentando, portanto, um grande potencial em aplicações que visam substituir ou complementar partes responsáveis por funções biológicas em seres humanos, como a visão. Ressalta-se aqui a retina, composta por diversas camadas celulares e cuja importante parcela de contribuição no processamento da informação visual ainda está sendo compreendida, envolvendo operações compatíveis com a realização por sistemas artificiais. Isso pode ser constatado no modelo proposto em (ZAGHLOUL, 2009), que integra elementos que representam operações de filtragem espaço-temporal, e em (GOLLISCH e MEISTER, 2010), onde são descritas algumas funções identificadas que são realizadas pelas células retinianas, como a detecção de movimento.

Entretanto, casos como os ilustrados anteriormente podem ser passíveis de uma restrição considerável de tempo ou energia, o que tende a aumentar a atratividade para realizar o processamento de forma analógica, que costuma ser mais rápido e eficiente, ainda que se acarrete em uma perda de precisão. Uma candidata bem adequada para tal tarefa é a rede neuronal celular (CNN - *cellular neural network*) analógica, cuja arquitetura original é proposta em (CHUA e YANG, 1988), onde células são dispostas em uma matriz e conectadas com suas vizinhas, cuja versatilidade possibilita a execução de uma ampla gama de operações em imagens, conforme retratado em (VADDI, BOGGAVARAPU, *et al.*, 2011), (HAO, JI e ZHOU, 2014) e (BOTOCA, 2014). Convém ressaltar que tanto a rede convolucional e a rede celular compartilham usualmente a mesma abreviação (CNN). No âmbito deste trabalho, todavia, este termo estará associado exclusivamente às redes neuronais celulares.

Seguindo uma abordagem com maior foco na redução de consumo, o trabalho iniciado em (SANTANA, FREIRE e CUNHA, 2012) propôs uma nova arquitetura para as sinapses de uma CNN baseada no modelo alternativo da célula tradicional denominado *Full Signal Range* (FSR), retratado em (ESPEJO, CARMONA, *et al.*, 1996) e que confere

uma maior simplicidade ao circuito, concebido em tecnologia CMOS. Além de admitir uma faixa contínua de valores para os coeficientes, a rede proposta em (SANTANA, FREIRE e CUNHA, 2012) aproveita o compartilhamento de blocos para reduzir não apenas o tamanho e a complexidade do sistema, como também o consumo e o descasamento entre seus elementos. Resultados adicionais ilustrados em (SANTANA, 2013) revelaram um bom desempenho no caso de funções simples de processamento de imagens monocromáticas bipolares. Por sua vez, (ANDRADE, SOUZA, *et al.*, 2015b) e (ANDRADE, 2015a), por meio de técnicas de treinamento, ampliaram a aplicabilidade da CNN de (SANTANA, 2013) não somente para funções bipolares mais complexas, como também para operações que trabalham em escala de cinza, explorando filtros espaciais. Todavia, tais testes envolveram uma rede com uma única camada, o que possivelmente dificultou uma reprodução bem fiel da filtragem obtida por meios convencionais, como a aplicação da transformada rápida de Fourier (FFT). Logo, visando a diminuição dos erros gerados pela resposta da CNN nestes casos, bem como expandir a verificação da funcionalidade dessa arquitetura com operações ainda mais complexas, torna-se necessário incrementar a capacidade de processamento da CNN. Duas alternativas imediatas para este fim são a ampliação da vizinhança das células e a formação de camadas com a sobreposição de múltiplas redes tradicionais. Enquanto o primeiro recurso permite de forma direta a aplicação de filtros espaciais com máscaras maiores (YANG, 2002), a segunda opção é mais atrativa do ponto de vista funcional, já que a princípio se trata de uma expansão mais significativa do ponto de vista estrutural e traz a vantagem de permitir também a aplicação de sucessivas operações em sequência, onde cada camada pode realizar uma etapa do processamento e enviar a resposta para a seguinte (YANG, NISHIO e USHIDA, 2003). Esta maior complexidade transfere-se, contudo, à configuração da rede, o que tende a dificultar o processo de estabelecer os valores dos seus parâmetros para uma determinada operação.

Com base nesta capacidade de expansão da CNN em camadas, em (YANG, NISHIO e USHIDA, 2002) e (YANG, NISHIO e USHIDA, 2003) foi proposta uma versão de CNN analógica de duas camadas. Uma característica distinta importante dessa arquitetura é a conexão entre as camadas nos dois sentidos, o que confere um acoplamento mútuo entre as camadas e permite a execução de operações que necessitariam a princípio de várias etapas a partir de uma rede com camada única, consequentemente acelerando o processo e potencialmente reduzindo o tamanho da implementação. Do mesmo modo, a utilização desta arquitetura pode ser vista como uma solução em potencial para tornar a

filtragem espacial executada pela rede mais fiel ao que se almeja, considerando aplicações com filtros projetados pelo método da frequência.

Um desafio que surge neste ponto é a necessidade de se obter os coeficientes sinápticos da CNN correspondentes à operação desejada. Muitas funções básicas de processamento de imagens, como a detecção de bordas ou a projeção de sombras, sendo em sua maioria do tipo bipolar, já têm esses parâmetros conhecidos, encontrados de forma analítica. Contudo, casos mais complexos, incluindo o dos filtros de imagens, podem demandar uma outra metodologia. Uma opção viável para a obtenção dos coeficientes em tais situações é o emprego de um algoritmo de treinamento, onde um determinado método numérico é aplicado iterativamente em conjunto com pares de entrada e saída que caracterizam a operação desejada, buscando convergir para uma solução cujos erros se restrinjam à faixa de tolerância admitida. Inúmeras abordagens para desempenhar esse processo têm sido desenvolvidas, dentre os quais podem-se citar exemplos determinísticos, como os baseados no gradiente do erro das saídas do sistema, e outros de caráter probabilísticos, dentre os quais se destacam as meta-heurísticas.

Considerando essa necessidade, o treinamento da CNN foi um dos elementos trabalhados em (ANDRADE, 2015a), utilizando uma técnica baseada na retropropagação do erro. Apesar de apresentar bons resultados para as funções tratadas, contando com uma grande velocidade de execução, testes posteriores com outras operações revelaram a dificuldade do algoritmo em solucionar certos casos com dinâmica mais complexa. Tal problema pode ser amplificado ao se estender o uso para redes de múltiplas camadas, exigindo a procura por alternativas com maior capacidade de convergência.

Seguindo essas ponderações, o presente trabalho é um prosseguimento natural de (ANDRADE, 2015a) e abrange dois elementos principais: a expansão da CNN desenvolvida em (SANTANA, 2013) para uma configuração em duas camadas e o aprimoramento da metodologia de treinamento para a aplicação das operações de processamento de imagens a serem tratadas, com o objetivo de analisar o desempenho dessa nova versão da rede a partir de simulações.

1.1 OBJETIVOS

Os principais objetivos deste trabalho de pesquisa são:

i. Expandir o circuito da CNN analógica baseada na célula em tecnologia CMOS proposta em (SANTANA, 2013) para uma versão em duas camadas;

ii. Prover métodos adequados para o treinamento da CNN de duas camadas, visando realização de funções complexas como a filtragem de imagens em escala de cinza;
iii. Aferir a confiabilidade da operação da CNN analógica de duas camadas por meio da simulação de diferentes funções de processamento de imagens.

São objetivos subjacentes: a condução de estudos comparativos sobre métodos de treinamento de CNN e a disponibilização de um compêndio sobre temas interdisciplinares relacionados a esta pesquisa.

1.2 CONTRIBUIÇÕES

Visando atingir os objetivos traçados, as seguintes contribuições originais foram produzidas:

i. Realização de uma versão pré-leiaute de uma CNN de duas camadas com acoplamento mútuo como a proposta por (YANG, NISHIO e USHIDA, 2002), utilizando a arquitetura de célula desenvolvida em (SANTANA, FREIRE e CUNHA, 2012) e aplicando modificações estruturais no circuito e ajustes empíricos na metodologia de aplicação de pesos para a melhoria de seu desempenho.
ii. Desenvolvimento de uma metodologia de treinamento para a rede em duas camadas, consistindo em:
    a. Uma versão estendida do Algoritmo de Centro de Massa descrito em (MIRZAI, CHENG e MOSCHYTZ, 1998) adaptada para a rede em duas camadas empregando os melhoramentos tratados em (ANDRADE, 2015a).
    b. Uma implementação mais robusta do Algoritmo Genético proposto em (KOZEK, ROSKA e CHUA, 1993), incorporando a capacidade de trabalho com uma representação real dos parâmetros do problema em conjunto com novos operadores genéticos compatíveis com esta forma de representação.
    c. Uma abordagem híbrida de treinamento formada pela combinação das duas técnicas citadas anteriormente.

iii. Análises comparativas dos métodos de treinamento empregados no trabalho, permitindo verificar o seu comportamento em diferentes situações.

iv. Avaliação do desempenho da CNN de duas camadas implementada a partir de simulações de funções bipolares e de filtragem espacial.

1.3 ESTRUTURA

Este trabalho está organizado em 5 capítulos. O capítulo 2 consiste numa revisita a conceitos fundamentais de várias disciplinas relacionadas a esta pesquisa. Além de introduzir a CNN e explicar seu funcionamento, estendendo a discussão para redes com duas camadas, aborda os temas de filtragem de imagens no domínio espacial e da frequência, algoritmos de treinamento de CNN, resposta em frequência da CNN e uma implementação de CNN analógica em tecnologia CMOS.

Os capítulos seguintes compreendem a contribuição original deste trabalho.

O capítulo 3 trata do treinamento de CNN e as metodologias empregadas, descrevendo os algoritmos desenvolvidos ou adaptados e detalhando o procedimento. Em seguida, são apresentadas análises comparativas entre as técnicas, acompanhada de resultados de exemplos de operações bipolares.

A realização da CNN de duas camadas é introduzida no capítulo 4, que se inicia com as considerações tomadas para sua simulação e segue com uma descrição dos aprimoramentos aplicados para compensação de imperfeições no processamento executado pelo circuito. Por fim, são exibidos resultados de exemplos de funções de duas camadas, subdivididos em operações bipolares e de filtragem de imagens.

O último capítulo inclui as conclusões da pesquisa desenvolvida e aborda os aspectos do trabalho visando possíveis desdobramentos.

2 FUNDAMENTAÇÃO TÉORICA

Para uma melhor compreensão das contribuições deste trabalho, neste capítulo será apresentada uma revisão teórica sobre temas diversos, porém inter-relacionados, que constituíram a base desta pesquisa: conceitos e definições associados à rede neuronal celular (CNN), incluindo o caso particular de duas camadas; descrições de dois métodos de treinamento de CNN aqui empregados: o algoritmo do centro de massa e o algoritmo genético; generalidades sobre a filtragem de imagens; a análise da operação de uma CNN no domínio da frequência; uma breve abordagem sobre a implementação analógica em tecnologia CMOS de uma CNN, utilizada nas simulações em nível de circuito aqui realizadas

2.1 REDE NEURONAL CELULAR

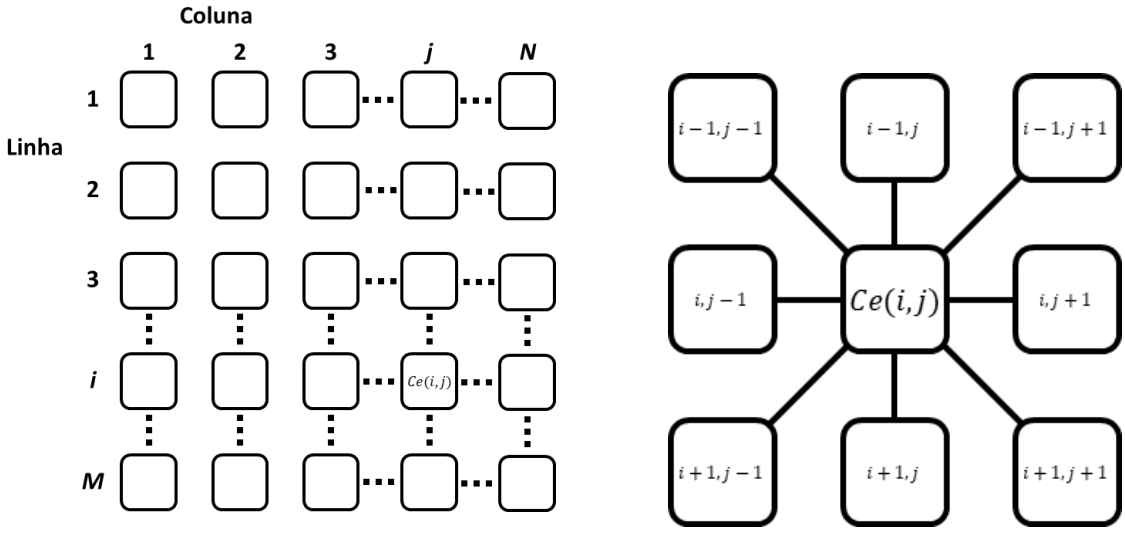
A rede neuronal celular (CNN) analógica é uma arquitetura computacional apresentada em (CHUA e YANG, 1988) que, como outras classes de redes neuronais artificiais (RNA), possui organização semelhante às redes de neurônios existentes em tecidos celulares e apresenta processamento paralelo. Além disso, sua dinâmica ocorre em tempo contínuo. A estrutura padrão de uma CNN é formada por células, unidades individuais de processamento distribuídas em uma matriz. A FIGURA 2.1.a ilustra a composição de uma rede de dimensões M por N, onde as células $Ce(i,j)$ estão identificadas por suas coordenadas cartesianas i e j, sendo estes números inteiros e positivos (CHUA e ROSKA, 2002).

Nesse arranjo, cada célula é conectada com as suas vizinhas, formando sua esfera de influência S_R, também denominado vizinhança, cujo raio R determina as distâncias máximas destas conexões. Sendo assim, uma rede de raio unitário, por exemplo, apresentará conexões apenas entre células adjacentes, incluindo as diagonais, como mostrado na FIGURA 2.1.b.

A dinâmica da CNN padrão é modelada pelas equações (2.1) e (2.2) e está representada também no diagrama de blocos da FIGURA 2.2 para o caso de uma célula.

$$\dot{x_{i,j}} = -x_{i,j} + \left[\sum_{Ce(k,l)\in S_R(i,j)} A(i,j;k,l)y_{k,l}\right] + \left[\sum_{Ce(k,l)\in S_R(i,j)} B(i,j;k,l)u_{k,l}\right] + z_{i,j} \quad (2.1)$$

$$y_{i,j} = f(x_{i,j}) = \frac{1}{2}|x_{i,j}+1| - \frac{1}{2}|x_{i,j}-1| \quad (2.2)$$



(a) (b)

FIGURA 2.1. (a) Estrutura básica de uma CNN. Adaptada de (CHUA e ROSKA, 2002). (b) Conexões entre uma célula e suas vizinhas para uma rede com $R = 1$.

As variáveis $x_{i,j}$, $y_{i,j}$, $u_{i,j}$ e $z_{i,j}$ são, respectivamente, o estado, a saída, a entrada e o limiar da célula $Ce(i,j)$, enquanto A e B são matrizes denominadas operadores sinápticos de realimentação e de entrada, nessa ordem.

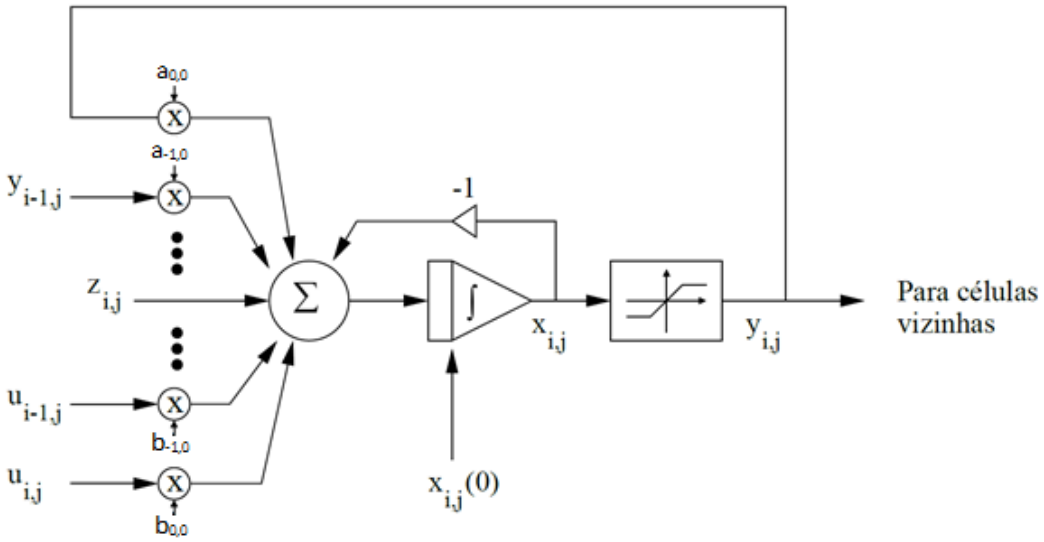


FIGURA 2.2. Diagrama de blocos de uma célula padrão. Extraído de (SANTANA, 2013).

A expressão (2.1) é denominada equação de estado, explicitando que a dinâmica da célula depende não somente de seus próprios sinais (entrada e saída), bem como dos sinais das células pertencentes à sua esfera de influência, $S_R(i,j)$. Os índices k e l correspondem às coordenadas destas células relativas à posição de $Ce(i,j)$. Já a equação

(2.2) estabelece a função de saída típica da célula, que age como limitadora do estado, sendo denominada não linearidade padrão, ilustrada na FIGURA 2.3. Enquanto o valor do estado pode assumir qualquer valor real, a aplicação desta função restringe a saída de cada célula ao intervalo $[-1, 1]$, que é a mesma faixa admitida para as entradas.

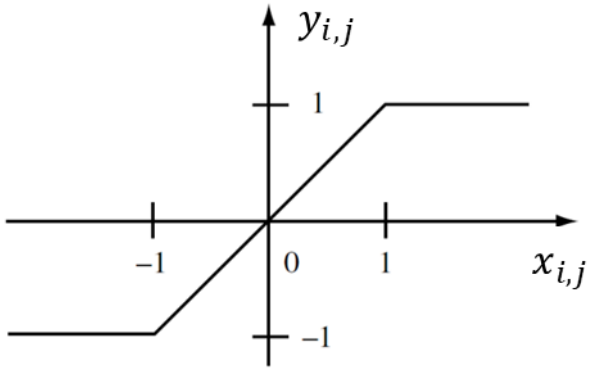


FIGURA 2.3. Não linearidade padrão. Extraída de (CHUA e ROSKA, 2002).

Dessa forma, a operação a ser executada pela CNN é determinada por um conjunto de coeficientes (ou pesos) composto pelos elementos das matrizes A e B e por $z_{i,j}$, sendo adequados para a maior parte das aplicações valores reais e invariantes no espaço e no tempo, condições que serão sempre assumidas a partir daqui. Sendo assim, tais operadores serão os mesmos para todas as células da rede e por simplicidade podem ser escritos dispensando-se os índices i e j. A depender da função configurada, o estado inicial pode ter ou não influência na resposta da rede. Em alguns casos o mesmo pode ser utilizado como o ponto de aplicação dos sinais a serem processados, ao invés dos terminais de entrada, dispensando o uso de B. Em outras ocasiões, onde se procura operar dois conjuntos de sinais simultaneamente, é possível utilizar como entradas da CNN os valores iniciais da variável $x_{i,j}$ (estado) e os valores da variável $u_{i,j}$ (de entrada propriamente dita).

Portanto, de acordo com a equação (2.1), a operação das sinapses em cada célula se resume, de forma geral, à multiplicação da saída $y_{k,l}$ de cada célula pertencente à vizinhança, bem como de sua entrada $u_{k,l}$, pelo operador sináptico correspondente à sua posição relativa (elementos de A e B, respectivamente), culminando no somatório de todos estes produtos e do valor de z. A integração deste resultado passa pela não-linearidade padrão para produzir o valor de saída. A resposta final será obtida quando a rede se estabilizar, no momento em que sua saída não apresente mais variações. Para

simplificar a representação dos coeficientes, podem ser definidos *templates* para toda a rede, associadas à função que será executada. A FIGURA 2.4 ilustra uma forma de representação dos *templates* para uma rede com vizinhança de raio unitário.

$a_{-1,-1}$	$a_{-1,0}$	$a_{-1,1}$
$a_{0,-1}$	$a_{0,0}$	$a_{0,1}$
$a_{1,-1}$	$a_{1,0}$	$a_{1,1}$

$b_{-1,-1}$	$b_{-1,0}$	$b_{-1,1}$
$b_{0,-1}$	$b_{0,0}$	$b_{0,1}$
$b_{1,-1}$	$b_{1,0}$	$b_{1,1}$

z

FIGURA 2.4. *Templates* para uma CNN com $R = 1$.

Outra consideração relevante são as condições de fronteira da rede, as quais afetam diretamente as células pertencentes às bordas, que a princípio teriam espaços vazios em sua esfera de influência. Para contornar isso são utilizados sinais que simulam a presença de células virtuais envolvendo a rede, mais especificamente conexões que representem suas entradas e saídas. Apesar de existirem múltiplas formas de se configurar esses elementos de contorno, neste trabalho foi considerado o método mais simples e que é aplicado na maioria dos casos, consistindo na escolha de valores fixos para os sinais reproduzidos por toda a fronteira, identificados como u_b e y_b. Esta fronteira virtual é retratada na FIGURA 2.5 para uma rede 4×4 com $R = 1$.

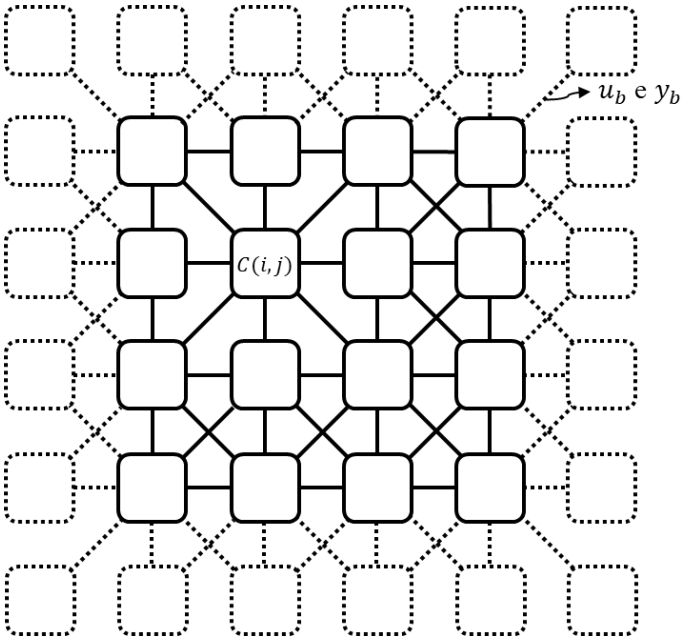


FIGURA 2.5. Esquema de conexão da fronteira para uma rede com $R = 1$. Os retângulos e as linhas tracejadas representam, respectivamente, as células de borda e os sinais fixos de entrada e saída.

Uma grande gama de funções executadas por CNN é classificada como bipolar (ou binária), isto é, trabalha apenas com entradas e saídas finais com valores -1 ou 1 (CHUA e ROSKA, 2002). Nessa situação, em geral a solução não é única, isto é, tanto a variação em uma faixa para cada parâmetro (elemento do *template*) quanto combinações diferentes de seus valores podem permitir a obtenção do mesmo resultado.

Uma função aplicada em uma CNN que não traz conexões dos sinais de saída entre células vizinhas, o que matematicamente significa que todos os coeficientes sinápticos de realimentação (elementos da matriz A) com exceção do central são nulos, é classificada como desacoplada (CHUA e ROSKA, 2002). Caso contrário, quando pelo menos um desses pesos não é nulo, a função é do tipo acoplada. Esta última categoria inclui exemplares caracterizados pela propagação de sinal pela rede de forma espacial, como preenchimento de buracos ou projeção de sombras. A FIGURA 2.6 descreve estas classificações, para uma CNN com $R = 1$.

Função desacoplada

0	0	0
0	$\hat{a}_C$	0
0	0	0

Sendo $\hat{a}_C \in \mathbb{R}$

Função acoplada

$a_{-1,-1}$	$a_{-1,0}$	$a_{-1,1}$
$a_{0,-1}$	$\hat{a}_C$	$a_{0,1}$
$a_{1,-1}$	$a_{1,0}$	$a_{1,1}$

Sendo $\hat{a}_C \in \mathbb{R}$ e pelo menos um dos coeficientes $a_{k,l}$ não nulo

FIGURA 2.6. Descrição dos tipos de função quanto ao acoplamento.

Em aplicações na área de processamento de imagens, vertente de maior destaque na utilização desse tipo de sistema, é comum associar os níveis mínimo (-1) e máximo (1) com o branco e o preto, respectivamente, em se tratando de imagens monocromáticas. Logo, considerando funções bipolares, as imagens envolvidas possuirão pixels apresentando sempre um destes dois níveis, o que permite a incorporação de operações binárias. Já nas situações não-bipolares, os níveis de sinal intermediários podem ser representados proporcionalmente por tons de cinza.

Visando uma arquitetura mais simples para as células da CNN, foi proposta em (RODRÍGUEZ-VÁZQUEZ, 1993) e analisada detalhadamente quanto à estabilidade em

(ESPEJO, CARMONA, *et al.*, 1996) uma variação do modelo tradicional chamada *Full Signal Range* (FSR), cuja principal diferença reside na limitação dos valores de estado nos mesmos níveis convencionados para a saída. Esta característica reduz a complexidade do projeto ao dispensar a necessidade do bloco limitador que implementa a função de saída (equação 2.2), contribuindo para a concepção de circuitos mais densos e robustos. Sendo assim, a dinâmica da célula $Ce(i,j)$ é modelada pela equação (2.3) (RODRÍGUEZ-VÁZQUEZ, 1993). A FIGURA 2.7 descreve o diagrama para este caso.

$$\dot{x_{i,j}} = gp(x_{i,j}) + \left[\sum_{Ce(k,l)\in S_R(i,j)} A_{FSR}(i,j;k,l)y_{k,l}\right] + \left[\sum_{Ce(k,l)\in S_R(i,j)} B(i,j;k,l)u_{k,l}\right] + z_{i,j} \quad (2.3)$$

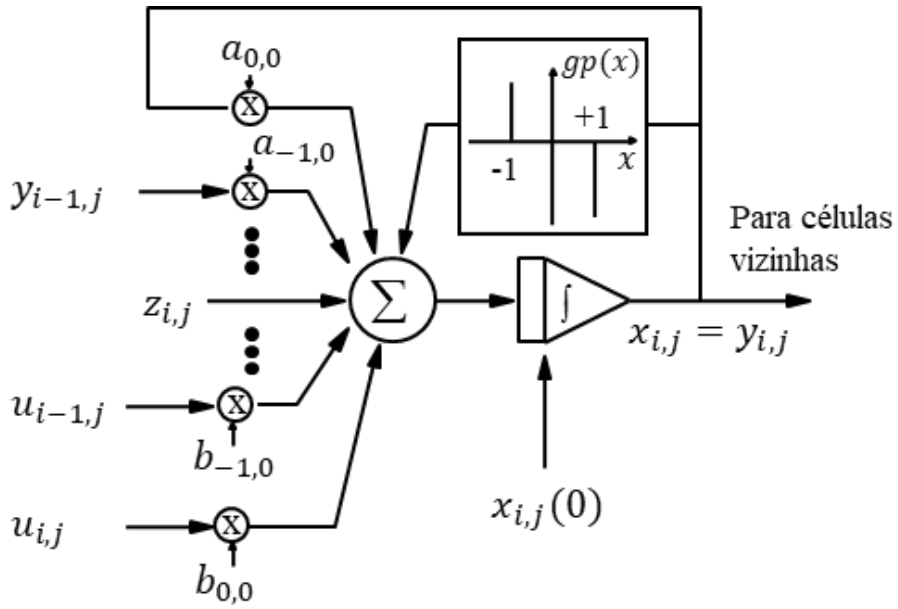


FIGURA 2.7. Diagrama de blocos de uma célula FSR. Extraído de (SANTANA, 2013).

A função de grampeamento $gp(x)$ substitui o bloco limitador e é definida como:

$$gp(x) = \lim_{m\to\infty} \begin{cases} -m(x+1), & x < -1 \\ 0, & -1 \le x \le 1 \\ -m(x-1), & x > 1 \end{cases} \quad (2.4)$$

Desta forma, nas células do tipo FSR, o estado da célula, que coincide com a variável de saída, estará limitado ao intervalo $[-1, 1]$, como as variáveis de entrada. Isto não acontece na célula tradicional, em cuja dinâmica as variáveis de estado podem atingir valores absolutos na ordem de 5 a 10 vezes maiores que as variáveis de saída. Por

conseguinte, a limitação exercida por $gp(x)$ nas células do tipo FSR contribui para reduzir a área e a potência do circuito, simplificando o seu projeto.

A partir da inspeção das equações (2.1) a (2.4), pode-se estabelecer uma relação entre a célula FSR e a tradicional, havendo uma equivalência entre suas dinâmicas quando a matriz A_{FSR} corresponde à matriz A ao se subtrair 1 do seu elemento central, como mostram a FIGURA 2.8 e a equação (2.5) para uma rede com vizinhança de raio unitário. Isso pode ser garantido, a princípio, apenas para situações em que o estado das células não ultrapasse o intervalo durante todo o processamento no caso da arquitetura padrão, já que o modelo FSR naturalmente impede esta condição. Na maioria das funções, os valores de estados nas células tradicionais podem se distanciar significativamente desta faixa. Contudo, a limitação exercida pela equação de saída das células tradicionais contribui para manter a equivalência descrita anteriormente no ponto de equilíbrio dos estados da rede, pois, mesmo apresentando trajetórias diferentes, as respostas dos dois casos coincidem em seu valor final. Este aspecto torna simples, portanto, o processo de adaptação dos coeficientes.

$a_{-1,-1}$	$a_{-1,0}$	$a_{-1,1}$
$a_{0,-1}$	$a_{0,0}-1$	$a_{0,1}$
$a_{1,-1}$	$a_{1,0}$	$a_{1,1}$

FIGURA 2.8. *Template* A_{FSR} em função dos elementos da matriz *A*, para uma rede com $R = 1$.

$$a_{FSR_{k,l,}} = \begin{cases} a_{k,l} - 1, & k = l = 0 \\ a_{k,l}\,, & caso\ contrário \end{cases} \tag{2.5}$$

2.1.1 Rede Neuronal Celular com Duas Camadas

A CNN descrita anteriormente constitui uma ferramenta bem versátil para o processamento de sinais, sobretudo nas aplicações envolvendo imagens, exibindo uma capacidade para realização de uma grande variedade de operações (CHUA e ROSKA, 2002). Contudo, com o passar do tempo, como costuma acontecer na evolução de qualquer sistema, o interesse pela exploração de funções mais complexas passou a crescer, incluindo exemplares que não podem ser realizados por uma rede simples de uma única vez. Tal entrave pode ser contornado tirando-se proveito de uma composição com múltiplas camadas da CNN, onde a operação é realizada em etapas e o fluxo do

processamento passa por cada camada sequencialmente. Outra possibilidade para este tipo de problema é o uso da Máquina Universal de CNN (CHUA e YANG, 1988), que pressupõe o uso de uma única CNN simples cujos coeficientes podem ser programados, de modo que em cada passo do processamento a rede é configurada para trabalhar de acordo com a operação desejada sobre os sinais resultantes da etapa anterior. Sendo assim, essa abordagem demanda a adição de blocos adicionais para armazenar os sinais e os coeficientes da rede.

Ambas metodologias descritas anteriormente podem ser consideradas formas de processamento serial e, portanto, tendem a ser mais lentas do que alternativas que tragam algum grau de paralelismo ao sistema. Nesse contexto, surgem as CNN de duas camadas com acoplamento mútuo, propostas em (YANG, NISHIO e USHIDA, 2002) e (YANG, NISHIO e USHIDA, 2003), denominadas aqui 2L-CNN e formadas pela sobreposição de CNN simples, referidas doravante como 1L-CNN. As conexões entre camadas, representadas por um novo conjunto de operadores sinápticos, podem existir nos dois sentidos, permitindo uma realimentação no processamento na forma de um acoplamento mútuo e possuem um alcance definido também pelo raio de vizinhança das células, considerado unitário neste desenvolvimento. As equações (2.6) e (2.7) definem a dinâmica dessa composição.

$$\dot{x_{1}}_{i,j} = -x_{1_{i,j}} + \left[\sum_{Ce(k,l,1)\in S_R(i,j,1)} A_1(i,j;k,l)y_{1_{k,l}}\right] + \left[\sum_{Ce(k,l,1)\in S_R(i,j,1)} B_1(i,j;k,l)u_{1_{k,l}}\right] + \left[\sum_{Ce(k,l,1)\in S_R(i,j,1)} C_1(i,j;k,l)y_{2_{k,l}}\right] + z_{1_{i,j}} \qquad (2.6.a)$$

$$\dot{x_{2}}_{i,j} = -x_{2_{i,j}} + \left[\sum_{Ce(k,l,2)\in S_R(i,j,2)} A_2(i,j;k,l)y_{2_{k,l}}\right] + \left[\sum_{Ce(k,l,2)\in S_R(i,j,2)} B_2(i,j;k,l)u_{2_{k,l}}\right] + \left[\sum_{Ce(k,l,2)\in S_R(i,j,2)} C_2(i,j;k,l)y_{1_{k,l}}\right] + z_{2_{i,j}} \qquad (2.6.b)$$

$$y_{1_{i,j}} = f\left(x_{1_{i,j}}\right) = \frac{1}{2}\left|x_{1_{i,j}} + 1\right| - \frac{1}{2}\left|x_{1_{i,j}} - 1\right| \qquad (2.7.a)$$

$$y_{2_{i,j}} = f\left(x_{2_{i,j}}\right) = \frac{1}{2}\left|x_{2_{i,j}} + 1\right| - \frac{1}{2}\left|x_{2_{i,j}} - 1\right| \qquad (2.7.b)$$

Os índices 1 e 2 referem-se às camadas da rede; $x_{m_{i,j}}$, $y_{m_{i,j}}$ e $u_{m_{i,j}}$ são, respectivamente, o estado, a saída e a entrada da célula $Ce(i,j,m)$, com m referindo-se ao índice da camada; A, B e C são os operadores sinápticos de realimentação, de entrada, e de acoplamento, respectivamente; $z_{m_{i,j}}$ é o limiar; e $S_R(i,j,m)$ é a região de vizinhança de $Ce(i,j,m)$. Pode-se notar que a única diferença nas equações em relação a uma CNN de uma camada é o terceiro termo das equações de estado (2.6), que se refere às conexões entre as camadas, ponderadas pelos elementos das matrizes C_1 e C_2, conforme descrito pelas expressões (2.6) e (2.7) e mostrado na FIGURA 2.9. O alcance das conexões entre as camadas é exemplificado na FIGURA 2.10.

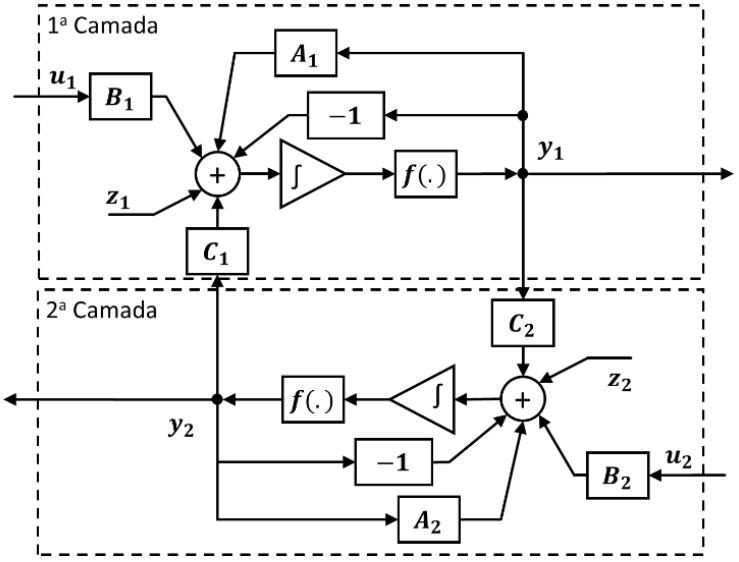


FIGURA 2.9. Diagrama de blocos simplificado da 2L-CNN. As conexões entre células vizinhas não estão representadas. Adaptado de (YANG, NISHIO e USHIDA, 2003).

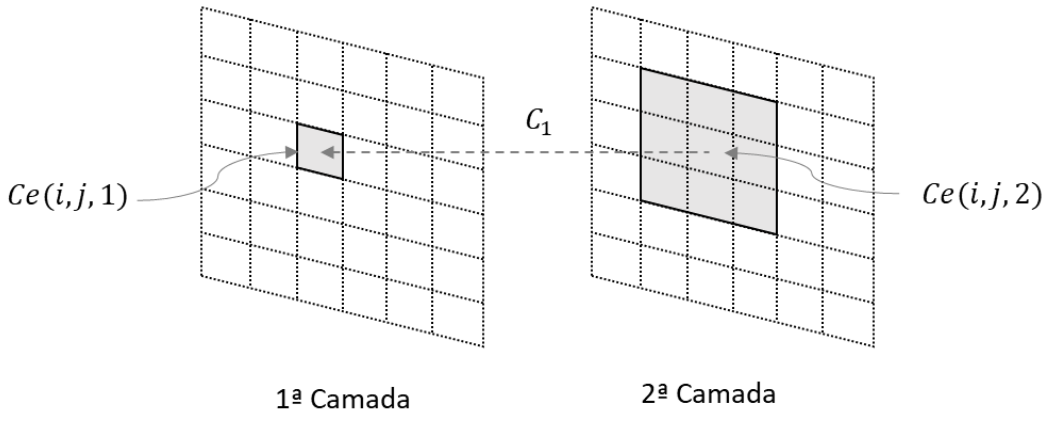
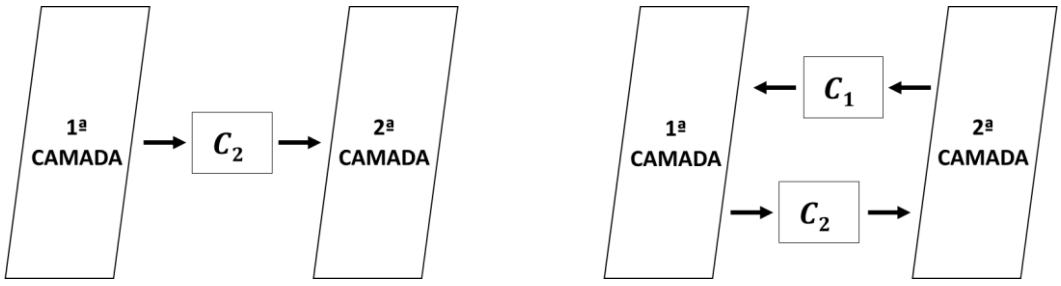


FIGURA 2.10. Exemplo de conexões entre camadas para $R = 1$.

Portanto, a escolha do uso de C_1 e C_2 define o comportamento da 2L-CNN: caso ambas sejam completamente nulas, obviamente se obterá duas CNN de uma camada independentes; se apenas uma delas for desconsiderada e existir apenas C_2, por exemplo, será formada uma composição de duas CNN em sequência, de forma semelhante ao descrito anteriormente, chamada aqui de 2LFF-CNN (*two-layers feedforward cellular neural network*); caso ambas matrizes tenham elementos não nulos, a rede apresentará uma realimentação entre as camadas e é este mecanismo que confere à rede simultaneamente maior versatilidade e uma capacidade de processamento paralelo entre as duas partes, permitindo uma execução mais eficiente de algumas funções ao se utilizar, por exemplo, cada camada para executar uma mesma operação em sentidos contrários ou realizar um revezamento do procedimento entre as partes, sendo de grande valia para casos com propagação de sinal. Esta configuração está identificada neste trabalho como 2LFB-CNN (*two-layers feedback cellular neural network*). Estas duas últimas possibilidades são representadas na FIGURA 2.11.



2LFF-CNN 2LFB-CNN

FIGURA 2.11. Esquemas de acoplamento entre camadas da 2L-CNN.

Um detalhe importante no que concerne à invariância espacial dos operadores considerada no trabalho é que esta propriedade, para a 2L-CNN, será estabelecida por camada. Sendo assim, podem-se representar as funções a serem aplicadas na rede de forma semelhante ao caso tradicional, com o *template* tendo o acréscimo de até duas matrizes (C_1 e C_2) com as mesmas dimensões que A e B. Adicionalmente, as condições de fronteira podem ser definidas para cada camada separadamente. Convém ressaltar também que a 2L-CNN pode ser realizada também com células do tipo FSR a partir do acréscimo da função de grampeamento e da remoção da função de saída, como indicado pelas relações (2.8) e pela FIGURA 2.12. Além disso, para a adaptação dos coeficientes, segue-se o mesmo princípio descrito anteriormente para a 1L-CNN.

$$\dot{x_{1_{i,j}}} = gp(x_{1_{i,j}}) + \left[\sum_{Ce(k,l,1)\in S_R(i,j,1)} A_1(i,j;k,l)x_{1_{k,l}}\right] + \left[\sum_{Ce(k,l,1)\in S_R(i,j,1)} B_1(i,j;k,l)u_{1_{k,l}}\right] + \left[\sum_{Ce(k,l,1)\in S_R(i,j,1)} C_1(i,j;k,l)x_{2_{k,l}}\right] + z_{1_{i,j}} \quad (2.8.a)$$

$$\dot{x_{2_{i,j}}} = gp(x_{2_{i,j}}) + \left[\sum_{Ce(k,l,2)\in S_R(i,j,2)} A_2(i,j;k,l)x_{2_{k,l}}\right] + \left[\sum_{Ce(k,l,2)\in S_R(i,j,2)} B_2(i,j;k,l)u_{2_{k,l}}\right] + \left[\sum_{Ce(k,l,2)\in S_R(i,j,2)} C_2(i,j;k,l)x_{1_{k,l}}\right] + z_{2_{i,j}} \quad (2.8.b)$$

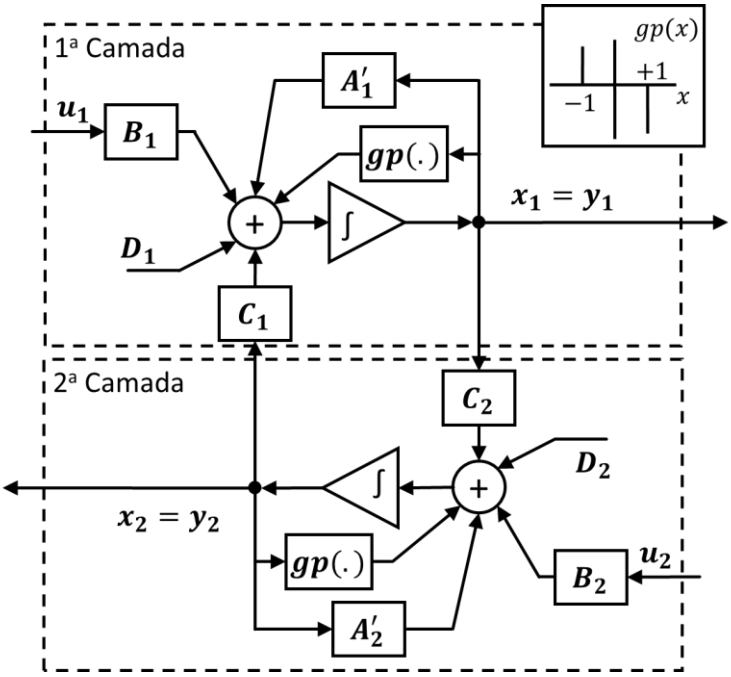


FIGURA 2.12. Diagrama de blocos da 2L-CNN. Adaptado de (YANG, NISHIO e USHIDA, 2003).

2.2 TREINAMENTO DE CNN

A CNN é dotada de uma estrutura que fornece uma ampla versatilidade. Contudo, para a sua aplicação uma das condições necessárias é a sua configuração, o que se traduz na escolha dos coeficientes utilizados. Em (NOSSEK, 1994), são citadas duas formas de se obter esses parâmetros: projeto ou aprendizado. A primeira alternativa envolve a descrição da função em termos de algumas leis dinâmicas locais. Todavia, a relação entrada/saída desejada (ou estado inicial/saída) pode ser complexa demais para permitir o enunciado de tais leis.

A outra forma de determinação dos coeficientes é o uso de técnicas de aprendizado ou treinamento, que são largamente utilizadas para outras categorias de redes neuronais. Estas abordagens baseiam-se no princípio de que, uma vez conhecido um conjunto de entradas e os resultados esperados para a função em questão, torna-se possível encontrar de forma iterativa os coeficientes da rede que reproduzem tal comportamento, através de algoritmos específicos. Logo, pode-se considerar este processo como um problema de otimização onde se busca minimizar o erro obtido pela CNN em cada iteração.

Seguindo essa lógica, um grande número de métodos foi desenvolvido contando com diversas inspirações. Os mais tradicionais utilizam o gradiente dos erros obtidos em cada iteração para indicar a direção que a busca pela solução deve seguir, dentre as quais se destacam aquelas fundamentadas na retropropagação (*backpropagation*) do erro. Estas vertentes costumam apresentar uma maior velocidade, ainda que possam ter problemas de convergência, principalmente pela sua sensibilidade a mínimos locais. Há ainda as meta-heurísticas, que incorporam conceitos originados de outras disciplinas, como os algoritmos genéticos e de enxame de partículas, inspirados na biologia, e o recozimento simulado (*Simulated annealing*), que reproduz fenômenos físicos. Tais métodos são usualmente de caráter estocástico, portanto não-determinísticos, tendendo a exigir mais esforço computacional, porém proporcionando uma maior capacidade de busca da solução global.

Alguns trabalhos surgiram posteriormente com versões de algoritmos adaptados para o caso específico de redes celulares (NOSSEK, 1994), (KOZEK, ROSKA e CHUA, 1993), (MIRZAI, CHENG e MOSCHYTZ, 1998), (LAI e WU, 2004), (MORENO-ARMENDARIZ, et al., 2006), (TANAKA, AOMORI, *et al.*, 2010). Essas questões foram consideradas em (ANDRADE, 2015), onde, visando uma maior simplicidade e velocidade, optou-se por implementar a metodologia proposta em (MIRZAI, CHENG e MOSCHYTZ, 1998), que trata do treinamento de CNN com base no Algoritmo do Centro de Massa (CMA*: Center of Mass Algorithm*), pertencente à classe de técnicas de retropropagação do erro e descrito na seção seguinte.

2.2.1 Algoritmo do Centro de Massa

O Algoritmo do Centro de Massa (CMA) é inspirado na técnica de aprendizado para RNA denominada retropropagação recorrente, cujo funcionamento engloba a variação dos parâmetros da rede guiada pelo gradiente do erro obtido a cada iteração (MIRZAI, CHENG e MOSCHYTZ, 1998). Como consequência, sua trajetória de busca

tende a ser bem rápida, podendo chegar a um mínimo com pouco esforço. Em contrapartida, esse comportamento o torna propenso a se prender em extremos locais, o que pode ser frequente em problemas complexos.

Uma característica do CMA é considerar apenas a resposta final da rede já estabilizada, ignorando a trajetória tomada durante o processamento. Além disso, o conceito de centro de massa proposto auxilia o treinamento para o caso de funções com uma dinâmica que dificulta a aplicação de algoritmos mais simples, como algumas funções acopladas, que envolvem propagação de sinal pela rede.

O método parte de um conjunto de coeficientes iniciais (A, B e z) e de pares de entrada e saída definidos para a função procurada, sendo possível também definir o estado inicial e as condições de fronteira. O resultado é um conjunto de coeficientes que permite a realização da operação. O desenvolvimento da técnica é feito para redes com vizinhança de raio unitário, onde os *templates* possuem dimensão 3x3.

O erro em cada célula é calculado a partir da equação (2.9):

$$e_{ij}[\kappa] = \frac{1}{2}\left(d_{ij} - y_{ij}[\kappa]\right) \tag{2.9}$$

onde e_{ij} e y_{ij} são o erro e a saída final da célula $Ce(i,j)$, respectivamente, e d_{ij} é a resposta desejada. O índice κ contabiliza as iterações. Considerando que a saída da CNN é limitada ao intervalo $[-1, 1]$, a divisão por 2 manterá o erro obtido a partir desta relação também nesta faixa.

A atualização dos coeficientes é realizada segundo (2.10):

$$a_{mn}[\kappa + 1] = a_{mn}[\kappa] + \eta \Delta a_{mn}[\kappa] \tag{2.10.a}$$

$$b_{mn}[\kappa + 1] = b_{mn}[\kappa] + \eta \Delta b_{mn}[\kappa] \tag{2.10.b}$$

$$z[\kappa + 1] = z[\kappa] + \eta \Delta z[\kappa] \tag{2.10.c}$$

sendo

$$\Delta a_{mn}[\kappa] = \begin{cases} 0\,, & se\ m = n = 2 \\ \dfrac{1}{MN} \displaystyle\sum_{1 \le i \le M, 1 \le j \le N} e_{ij}[\kappa] y_{i+m-2\ j+n-2}[\kappa]\,, & caso\ contrário \end{cases} \tag{2.11.a}$$

$$\Delta b_{mn}[\kappa] = \frac{1}{MN} \sum_{1 \le i \le M, 1 \le j \le N} e_{ij}[\kappa] u_{i+m-2\ j+n-2}[\kappa] \qquad (2.11.b)$$

$$\Delta z[\kappa] = \frac{1}{MN} \sum_{1 \le i \le M, 1 \le j \le N} e_{ij}[\kappa] \qquad (2.11.c)$$

onde $m, n \in \{1,2,3\}$, $\eta > 0$ é a taxa de aprendizado e M e N são, respectivamente, o número de linhas e colunas da CNN. Uma consideração importante para essas relações é a escolha do valor do elemento central da matriz A, a_{22}, que é mantido fixo durante o treinamento e cumpre o papel de referencial para os demais. Visando aplicações bipolares, é interessante que este operador seja maior ou igual a 1, já que se trata de uma condição predominante neste contexto, caso contrário, poderia haver uma inconsistência no sentido da atualização para a minimização do erro. Sendo assim, a cada passo κ a mudança de cada parâmetro dos operadores sinápticos é obtida a partir dos produtos entre o erro de cada célula e o sinal da sua vizinha com posição relativa correspondente a este parâmetro (para a_{12}, por exemplo, as multiplicações ocorrerão entre os erros das células e a saída da célula adjacente localizada ao norte). Essas operações são então somadas para incluir as contribuições de toda a rede. Já o limiar é variado de acordo com o acúmulo dos erros das células.

A atualização dos coeficientes em (2.10) pode ser explicada de forma intuitiva: uma célula contribuirá nas somas quando o erro em sua resposta for diferente de zero. Por exemplo, considerando isoladamente o caso de uma única célula $Ce(i,j)$ em uma função bipolar, se o valor de y_{ij} for -1, mas o valor desejado d_{ij} for 1, o erro terá o valor máximo de 1 e, para reduzi-lo, o algoritmo aplicará as relações em (2.11) visando o aumento do resultado das sinapses envolvidas, seja aumentando, proporcionalmente ao fator η, os coeficientes a_{mn} correspondentes às células vizinhas que apresentem $y_{mn} = 1$, seja diminuindo os coeficientes a_{mn} referentes às células com $y_{mn} = -1$. A realização do somatório para todos os valores de i de 1 a M e para todos os valores de j de 1 a N e a divisão pelo fator MN levam à média das contribuições fornecidas pela realização dessa análise em todas as células da rede, que por sua vez é aplicada na atualização ao final de cada passo.

Pode-se observar, portanto, que este método realiza o treinamento em lotes, no sentido em que os coeficientes são alterados apenas após a aplicação de (2.11) em todas

as células, enquanto outras técnicas podem executar essa atualização em fluxo contínuo, isto é, após o tratamento em cada célula.

Visando melhorar o desempenho do treinamento, em (MIRZAI, CHENG e MOSCHYTZ, 1998) são propostos ainda alguns incrementos na metodologia, no que concerne às matrizes de operadores sinápticos. A primeira refere-se ao operador *B*, sugerindo a fixação prévia do seu formato de acordo com a natureza da função desejada, considerando as possíveis direções e simetrias atreladas ao seu funcionamento. Tomando como exemplo a situação em que a matriz tem o modelo descrito em (2.12), a atualização será feita de acordo com (2.13).

$$B = \begin{bmatrix} 0 & b & 0 \\ b & b_c & b \\ 0 & b & 0 \end{bmatrix} \tag{2.12}$$

$$B = \begin{bmatrix} 0 & \Delta b[\kappa] & 0 \\ \Delta b[\kappa] & \Delta b_{22}[\kappa] & \Delta b[\kappa] \\ 0 & \Delta b[\kappa] & 0 \end{bmatrix} \tag{2.13}$$

onde a grandeza $\Delta b[\kappa]$ é dada por:

$$\Delta b[\kappa] = \frac{\Delta b_{12}[\kappa] + \Delta b_{21}[\kappa] + \Delta b_{23}[\kappa] + \Delta b_{32}[\kappa]}{4} \tag{2.14}$$

A segunda modificação procede do fato, citado anteriormente, de que este algoritmo não considera a evolução temporal para o cálculo do erro. Isto dificulta a implementação de certos tipos de funções com propagação de sinal, como a projeção de sombras. Para compensar esta limitação, durante o tratamento do operador *A*, realiza-se primeiramente a decomposição em três partes, mostradas em (2.15): uma componente simétrica (em relação ao centro), A_s, uma anti-simétrica, A_a, e A_c.

$$A_s = \frac{1}{2}\begin{bmatrix} a_{11} + a_{33} & a_{12} + a_{32} & a_{13} + a_{31} \\ a_{21} + a_{23} & 0 & a_{21} + a_{23} \\ a_{13} + a_{31} & a_{12} + a_{32} & a_{11} + a_{33} \end{bmatrix} \tag{2.15.a}$$

$$A_a = \frac{1}{2}\begin{bmatrix} a_{11} - a_{33} & a_{12} - a_{32} & a_{13} - a_{31} \\ a_{21} - a_{23} & 0 & a_{23} - a_{21} \\ a_{31} - a_{13} & a_{32} - a_{12} & a_{33} - a_{11} \end{bmatrix} \tag{2.15.b}$$

$$A_c = A - A_s - A_a \tag{2.15.c}$$

As duas primeiras matrizes podem ser relacionadas a comportamentos distintos do funcionamento da CNN: a componente simétrica corresponde às dinâmicas locais e de difusão da função, enquanto a anti-simétrica lida com as dinâmicas globais e de propagação. Ambas são atualizadas de maneiras distintas, tirando proveito dessas características. A componente A_c permanece fixa durante o treinamento. No final de cada passo, as três matrizes são somadas para obter o novo operador A. De acordo com a função que se espera reproduzir, pode-se optar por utilizar apenas uma das componentes (A_s ou A_a), o que é suficiente para muitos casos conhecidos, sendo a sua maioria dotada de uma dinâmica simétrica.

A atualização de A_s é realizada conforme (2.16), onde o índice κ foi omitido na matriz ΔA_S para simplificação. Nota-se que este procedimento é baseado nas operações descritas em (2.11.a), introduzindo, entretanto, a soma da matriz das variações Δa_{mn} com sua versão rotacionada por 180º, com o intuito de manter a simetria.

$$A_s[\kappa + 1] = A_s[\kappa] + \eta \Delta A_s[\kappa] \tag{2.16.a}$$

$$\Delta A_s = \frac{1}{2}\begin{bmatrix} \Delta a_{11} + \Delta a_{33} & \Delta a_{12} + \Delta a_{32} & \Delta a_{13} + \Delta a_{31} \\ \Delta a_{21} + \Delta a_{23} & 0 & \Delta a_{21} + \Delta a_{23} \\ \Delta a_{13} + \Delta a_{31} & \Delta a_{12} + \Delta a_{32} & \Delta a_{11} + \Delta a_{33} \end{bmatrix} \tag{2.16.b}$$

Por sua vez, a componente A_a é atualizada utilizando a noção de centro de massa, aproveitando a organização da rede em duas dimensões. Busca-se neste caso eliminar a diferença entre as coordenadas do centro de massa da saída da rede e o da saída desejada, aproximando-os a cada iteração. A princípio, atribui-se a cada célula $Ce(i,j)$ um valor de massa m_{ij} dependente do valor na sua saída, de acordo com a expressão (2.17):

$$m_{ij} = \frac{1 + y_{ij}}{2} \tag{2.17}$$

Além disso, como $-1 \leq y_{ij} \leq 1$, tem-se que $0 \leq m_{ij} \leq 1$. Assim, o cálculo do centro de massa em relação a um eixo l é feito segundo a equação (2.18):

$$r_l = \frac{1}{M_t} \sum_{1 \leq i \leq M, 1 \leq j \leq N} d(l)_{ij} m_{ij} \tag{2.18}$$

sendo $d(l)_{ij}$ a distância entre a célula $Ce(i,j)$ e o eixo l. A variável M_t é a soma das massas de todas as células da rede.

Para cada elemento de A_a, o centro de massa é calculado em relação a um eixo dentre quatro possibilidades, ilustradas na FIGURA 2.13, de acordo com a sua posição na matriz.

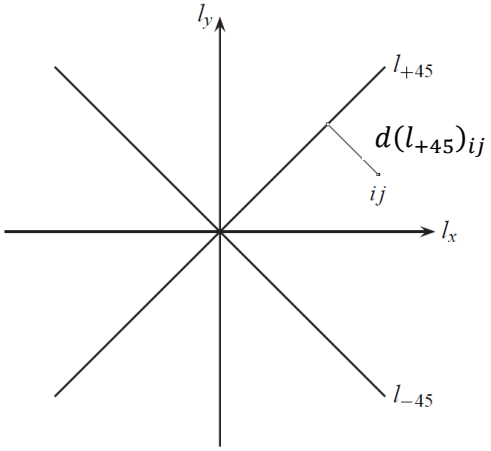


FIGURA 2.13. Eixos utilizados para o cálculo do centro de massa. Adaptada de (MIRZAI, CHENG e MOSCHYTZ, 1998).

Escrevendo as distâncias em função do par cartesiano (i, j), chegamos às expressões (2.19) para cada eixo:

$$r_{lx} = \frac{1}{M_t} \sum_{1 \le i \le M, 1 \le j \le N} j m_{ij} \qquad (2.19.a)$$

$$r_{ly} = \frac{1}{M_t} \sum_{1 \le i \le M, 1 \le j \le N} i m_{ij} \qquad (2.19.b)$$

$$r_{l+45} = \frac{1}{M_t} \frac{\sqrt{2}}{2} \sum_{1 \le i \le M, 1 \le j \le N} |i - j| m_{ij} \qquad (2.19.c)$$

$$r_{l-45} = \frac{1}{M_t} \frac{\sqrt{2}}{2} \sum_{1 \le i \le M, 1 \le j \le N} (i + j) m_{ij} \qquad (2.19.d)$$

A partir das expressões (2.19) pode-se calcular as diferenças entre o centro de massa da resposta obtida pela rede (r_l^y) e da resposta desejada (r_l^d), utilizando (2.20):

$$\Delta_{lx} = r_{lx}^d[\kappa] - r_x^y[\kappa] \qquad (2.20.a)$$

$$\Delta_{ly} = r_{ly}^d[\kappa] - r_{ly}^y[\kappa] \qquad (2.20.b)$$

$$\Delta_{l+45} = r^{d}_{l+45}[\kappa] - r^{y}_{+45}[\kappa] \qquad (2.20.c)$$

$$\Delta_{l-45} = r^{d}_{l-45}[\kappa] - r^{y}_{-45}[\kappa] \qquad (2.20.d)$$

E, finalmente, A_a pode ser atualizada de acordo com (2.21):

$$A_a[\kappa + 1] = A_a[\kappa] + \eta \Delta A_a[\kappa] \qquad (2.21.a)$$

$$\Delta A_a[\kappa] = \begin{bmatrix} \Delta_{l-45} & \Delta_{ly} & \Delta_{l+45} \\ \Delta_{lx} & 0 & -\Delta_{lx} \\ -\Delta_{l+45} & -\Delta_{ly} & -\Delta_{l-45} \end{bmatrix} \qquad (2.21.b)$$

Ainda em (MIRZAI, CHENG e MOSCHYTZ, 1998), são feitas verificações do método com o treinamento de uma CNN convencional para algumas funções bipolares simples, apresentando um bom desempenho. Contudo, para aferir a confiabilidade de tal metodologia de forma mais abrangente, o CMA foi aplicado em (ANDRADE, SOUZA, *et al.*, 2015b) e (ANDRADE, 2015a) para o treinamento de uma CNN FSR com operações de processamento de imagens de caráter mais complexo, tendo em vista a obtenção de coeficientes a serem utilizados em simulações da rede desenvolvida em (SANTANA, 2013).

2.2.2 Algoritmo Genético

O GA (*genetic algorithm*), descrito em (GOLDBERG, 2008), é uma classe de métodos de otimização pertencente ao ramo da computação evolucionária, sendo, portanto, inspirado em mecanismos conhecidos da teoria da evolução, especificamente na genética e na seleção natural. A definição clássica engloba algumas características:

i. Utiliza uma codificação binária do conjunto de parâmetros a serem otimizados.
ii. Realiza a busca a partir de uma população de pontos, de forma simultânea.
iii. Considera apenas a função de custo na otimização, ignorando outras informações do problema, como gradientes.
iv. Obedece a regras de transição probabilísticas, geralmente aplicadas via operadores baseados em processos genéticos.

Esses aspectos, especialmente o terceiro, conferem ao GA uma expressiva habilidade para se chegar a uma solução global, favorecendo o seu uso para problemas com dinâmicas complexas, incluindo a presença forte de mínimos locais. Além disso,

apesar de apresentar um caráter aleatório em algumas de suas etapas, o fato de o método aproveitar informações acumuladas durante as iterações o ajuda a ser mais rápido do que uma técnica puramente estocástica.

A FIGURA 2.14 mostra um fluxograma com as principais etapas de um algoritmo deste tipo. Basicamente, seu funcionamento pode ser descrito da seguinte forma:

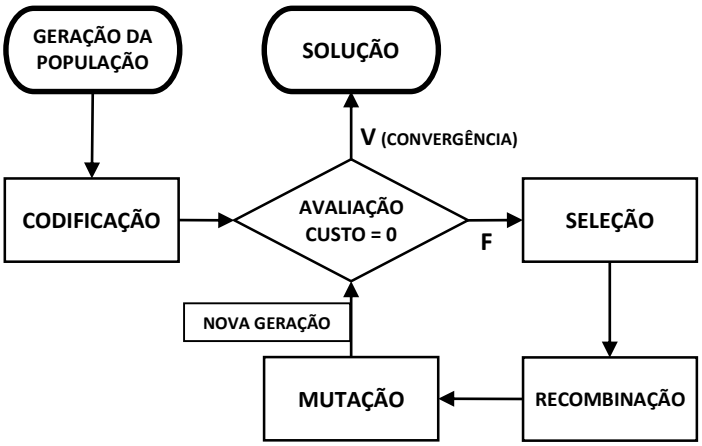


FIGURA 2.14. Fluxograma de um algoritmo genético.

i. Como ponto de partida, é gerada uma população inicial de indivíduos ou cromossomos, cada um contendo um conjunto dos parâmetros do problema;
ii. Os indivíduos passam por uma codificação para gerar os códigos binários correspondentes;
iii. Os indivíduos são avaliados por uma função que atua sobre os custos, produzindo um valor de aptidão para cada indivíduo;
iv. Caso não haja convergência, ou seja, se nenhum dos indivíduos for uma solução aceitável, serão selecionados dentre eles pares para recombinação, normalmente levando em conta seu grau de aptidão, com indivíduos mais aptos tendo maior chance de seleção;
v. Na recombinação, um operador é responsável pela troca de conteúdo genético, gerando descendentes.
vi. E, por último, eles podem passar por um operador de mutação para alterações adicionais no seu código.

Após esta última etapa, repete-se a avaliação com a nova geração e este ciclo recomeça até um dos critérios de parada ser atingido. Espera-se que haja uma evolução em relação ao desempenho dos parâmetros a serem otimizados, representados pelos

indivíduos, e que surja em algum momento um exemplar que traga uma solução que satisfaça o problema, encerrando o processo.

As variações dos cromossomos são realizadas por operadores genéticos, como os citados nos itens v e vi, e normalmente funcionam de forma probabilística. Apesar disso, o algoritmo trabalha de forma que, com o passar das gerações, os cromossomos mais aptos terão mais chances de permanecer no conjunto; blocos do cromossomo que trazem características favoráveis também tendem a ter uma presença maior.

Adicionalmente, a possibilidade de variação dos mecanismos de codificação e avaliação fornece uma grande versatilidade ao GA, à medida que tais operações podem ser implantadas tendo em vista o caráter do problema, culminando em um algoritmo que pode ser melhor adaptado a algumas de suas nuances, como por exemplo o número de parâmetros envolvidos ou as relações entre eles.

2.2.2.1 Implementação para Treinamento de CNN

Em (KOZEK, ROSKA e CHUA, 1993), é apresentada uma versão de GA formulada especificamente para CNN, seguindo o fluxograma da FIGURA 2.14. São incluídas diferentes alternativas para algumas etapas do algoritmo, como os operadores, sendo que cada opção pode trazer vantagens próprias para certas caraterísticas do problema em questão.

A equação (2.22) ilustra a função de custo estabelecida para cada indivíduo,

$$ct(p) = \sum_{i=1}^{M} \sum_{j=1}^{N} \left(d_{ij} - y_{ij}\right)^2 \tag{2.22}$$

onde $ct(p)$ é o custo associado ao indivíduo p, d_{ij} e y_{ij} são, respectivamente, a resposta desejada e saída final da célula $Ce(i,j)$, esta última proveniente da execução da CNN utilizando os parâmetros correspondentes. Logo, a subtração existente na expressão equivale ao erro da célula e, consequentemente, o custo estará associado ao seu valor quadrático.

A codificação converte o conjunto de coeficientes de cada indivíduo em vetores de bits. Um detalhe que convém ser observado é que o tamanho do vetor binário dependerá de duas características: o número de coeficientes a serem treinados e a precisão especificada (que influencia o número de bits por parâmetro). E esse comprimento pode influenciar de forma significativa o desempenho do algoritmo, dependendo da forma com

que se realiza as etapas de seu fluxo. Nesse contexto, são propostas três formas de codificação, ilustradas na TABELA 2.1:

i. Codificação padrão (*Standard coding*): os coeficientes na base binária são concatenados em sequência. Uma limitação desta variante é a tendência a uma redução de desempenho à medida que o comprimento do cromossomo aumenta, sendo este efeito decorrente da amplificação da chance de haver mudanças bruscas em apenas uma parte dos coeficientes, o que por sua vez pode inviabilizar uma busca mais concentrada.

ii. Codificação aprimorada (*Enhanced coding*): os bits de mesma ordem em cada parâmetro são emparelhados e posteriormente concatenados. Apesar de ser mais complexo que o método anterior, o mesmo é menos sensível à variação do tamanho do cromossomo, já que para o caso da CNN o sinal e a razão dos valores dos coeficientes são mais importantes do que a sua magnitude.

iii. Reordenamento por inversão (*Reordering by inversion*): depois da avaliação, cada cromossomo tem um trecho reordenado de maneira inversa. Um outro vetor de inteiros auxilia no armazenamento da sua ordem original. Assim, o indivíduo passa pelas operações de recombinação e mutação com essa alteração, o que gera mais uma fonte de variabilidade em relação ao método padrão, sendo a ordem restaurada posteriormente, durante a decodificação.

TABELA 2.1. Exemplo de aplicação dos métodos de codificação.

Coeficientes (Vetor binário)		$p = [p_1\ p_2\ p_3\ p_4] \quad s = [s_1\ s_2\ s_3\ s_4] \quad t = [t_1\ t_2\ t_3\ t_4]$
Codificação	Padrão:	$[p_1\ p_2\ p_3\ p_4 \quad s_1\ s_2\ s_3\ s_4 \quad t_1\ t_2\ t_3\ t_4]$
	Aprimorada:	$[p_1\ s_1\ t_1 \quad p_2\ s_2\ t_2 \quad p_3\ s_3\ t_3 \quad p_4\ s_4\ t_4]$
	Inversão:	$\begin{matrix} & p_1 & p_2 & & s_3 & s_2 & s_1 & p_4 & & p_3 & s_4 & t_1 & t_2 & t_3 & t_4 \\ Índices: & 1 & 2 & & \underbrace{7 & 6 & 5 & 4}_{\text{Fragmento invertido}} & & 3 & 8 & 9 & 10 & 11 & 12 \end{matrix}$

A avaliação da população é realizada a partir dos resultados da aplicação dos coeficientes de cada indivíduo em um modelo da CNN, onde se infere o erro da sua resposta. A função custo é então calculada e seu valor passa por um mapeamento que gera

a aptidão (*fitness*) ft do cromossomo, a qual deve ser maximizada pelo algoritmo. Três formas de avaliação são propostas em (KOZEK, ROSKA e CHUA, 1993):

i. Mapeamento direto (*Direct mapping*): transformação elementar do custo a ser minimizado para um valor de aptidão.

O cálculo é feito pela equação (2.23):

$$ft_{ind} = maxdiff - ct(p_{ind}) \tag{2.23}$$

sendo ind o índice do indivíduo, $maxdiff$ a soma dos valores das saídas das células no caso da maior diferença possível entre a imagem desejada e a obtida pela CNN. Como a saída de cada célula tem valor no intervalo $[-1, 1]$, o erro associado está contido na faixa $[-2, 2]$. Logo, tem-se para uma rede de dimensões MxN:

$$maxdiff = 4 \times M \times N \tag{2.24}$$

O gráfico da FIGURA 2.15 ilustra esse mapeamento. Uma limitação dessa técnica é que, conforme o treinamento evolui, as diferenças entre as aptidões dos indivíduos tendem a decrescer, dificultando o sucesso da etapa de seleção. As duas opções seguintes visam justamente contornar esse aspecto.

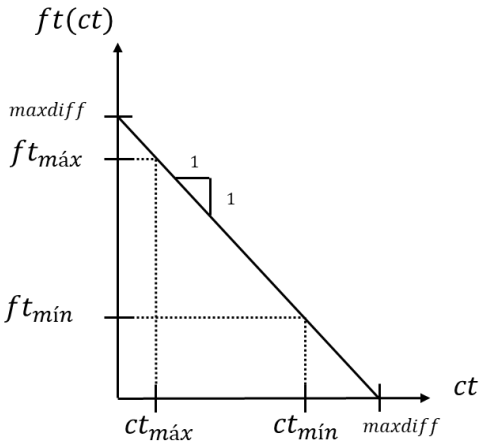


FIGURA 2.15. Avaliação por mapeamento direto. Adaptada de (KOZEK, ROSKA e CHUA, 1993).

ii. Janelamento (*Windowing*): Um valor mínimo ft_{min} é associado ao pior indivíduo daquela geração, servindo como referência. A partir disso, cada um dos membros restantes é creditado com um valor de aptidão

proporcional à diferença entre o seu custo e o do pior indivíduo. A FIGURA 2.16 mostra um exemplo do processo.

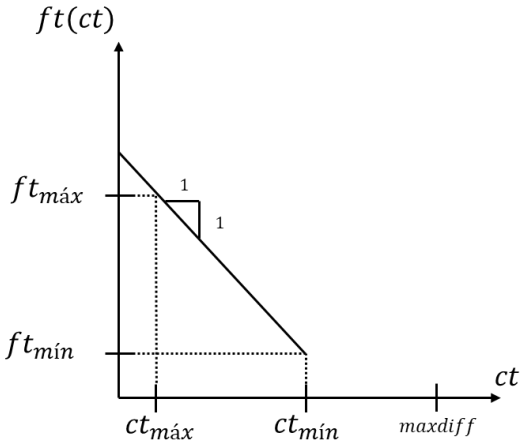


FIGURA 2.16. Avaliação por janelamento. Adaptada de (KOZEK, ROSKA e CHUA, 1993).

iii. Escalonamento linear (*Linear scaling*): Primeiramente, a aptidão bruta é calculada usando o mapeamento direto. Em seguida, uma função linear mapeia a aptidão bruta em um novo valor de aptidão, tal que o custo médio da população é mapeado na aptidão média e um valor mínimo de aptidão ft_{min} é atribuído ao indivíduo com maior custo, conforme retratado na FIGURA 2.17.

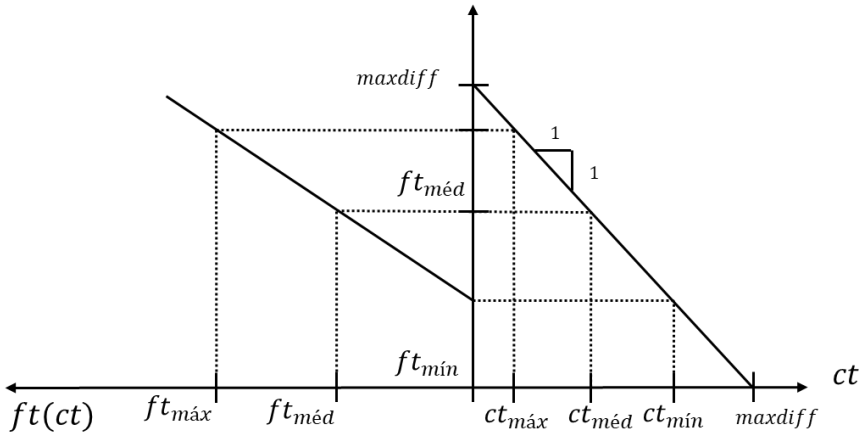


FIGURA 2.17. Avaliação por escalonamento linear. Adaptada de (KOZEK, ROSKA e CHUA, 1993).

A composição de novas gerações é formada pelo resultado da sequência de operadores de seleção, recombinação e mutação, mantendo o número de indivíduos np. A seleção é responsável por fornecer um conjunto com np indivíduos, com a possibilidade de repetições. Tal procedimento consiste em uma variação do mecanismo

clássico: indivíduos acima da média são pré-selecionados automaticamente e seus valores de aptidão são decrescidos do valor de aptidão médio. Para completar as demais vagas, todos os membros da população passam por um sorteio onde suas probabilidades de escolha são proporcionais aos novos valores de aptidão. Logo, a probabilidade de seleção de um indivíduo in nesta etapa é dada pela equação (2.25):

$$Ps_{in} = \frac{ft_{in}}{\sum_{k=1}^{np} ft_k} \tag{2.25}$$

Cada par de indivíduos selecionados irá passar em seguida pela recombinação, onde seu conteúdo genético será trocado, resultando em um par de descendentes. Em (KOZEK, ROSKA e CHUA, 1993) são citadas três alternativas, exemplificadas na FIGURA 2.18:

i. Cruzamento de um ponto (*One-point crossover*): Uma posição é selecionada segundo uma probabilidade uniforme ao longo do comprimento dos cromossomos. Os bits depois deste ponto são então trocados entre os dois indivíduos.

ii. Cruzamento de dois pontos (*Two-point crossover*): Segue o mesmo princípio do método anterior, com a diferença que dois pontos são selecionados. Os bits entre estes pontos são então trocados entre os dois indivíduos. Este método tem a vantagem de possibilitar recombinações mais variadas, logo, tende a ter melhores resultados que a versão de um ponto, à medida que o comprimento do cromossomo aumenta.

iii. Cruzamento aleatório (*Random crossover*): Este operador recombina dois cromossomos de acordo com uma sequência binária aleatória. Nos locais em que o bit dessa cadeia é 1, os bits correspondentes dos indivíduos são trocados; caso seja 0, não há troca. A característica mais aleatória torna o treinamento mais imprevisível, contudo o torna insensível ao tamanho do vetor e pode ser uma alternativa útil para casos onde os demais tipos de recombinação mostram um baixo desempenho.

Quanto à mutação, utiliza-se o procedimento mais comum em algoritmos genéticos, onde o operador passa por cada bit dos indivíduos e pode inverter o seu valor de acordo com uma chance percentual pré-determinada.

Um cuidado importante para não haver redundância é evitar a existência de indivíduos idênticos na nova geração, bem como de descendentes iguais a seus pais. Isso

pode ser conseguido tomando-se algumas providências. Inicialmente, quando alguma dessas condições ocorre, pode-se trocar os pontos ou o vetor de recombinação. Se isso não for suficiente, troca-se aleatoriamente um dos pais. Como último recurso, recorre-se à aplicação de mutação em bits aleatórios até eliminar as repetições.

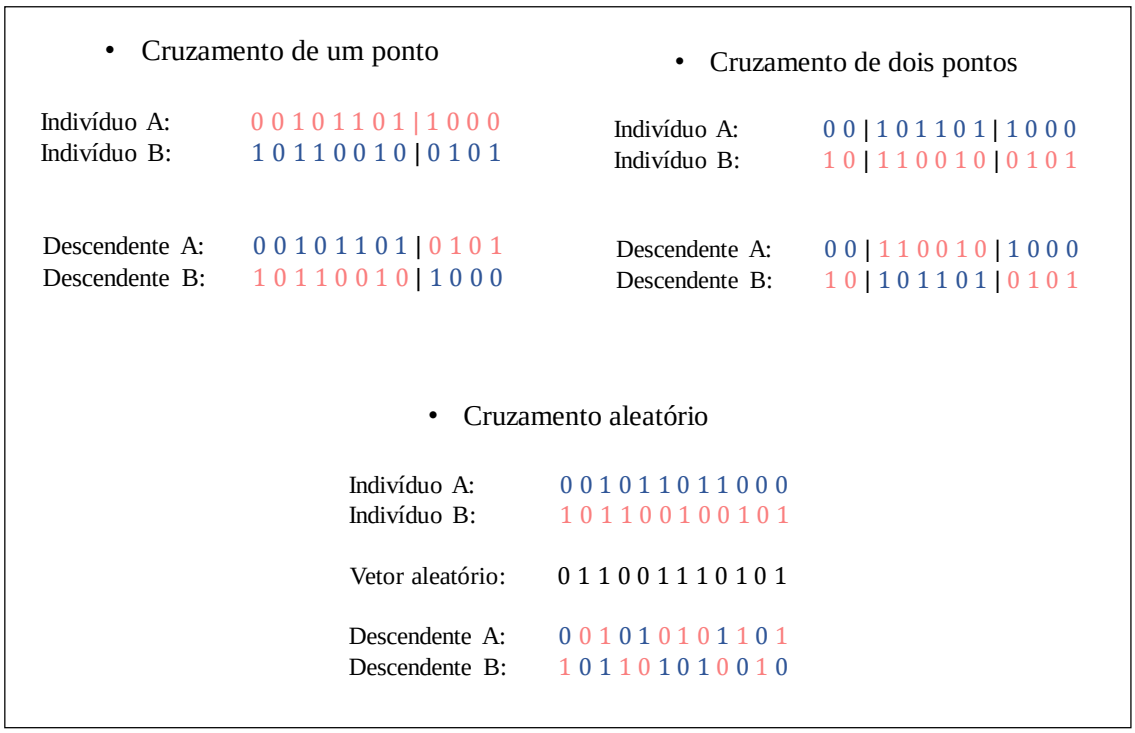


FIGURA 2.18. Formas de recombinação. Adaptada de (KOZEK, ROSKA e CHUA, 1993).

Um último conceito que pode ser aproveitado nessa metodologia é o elitismo, que consiste em reservar um determinado número de vagas da próxima geração para os cromossomos melhor avaliados, que são transferidos diretamente ignorando os operadores genéticos. Como consequência, a quantidade de descendentes produzidos pelo fluxo convencional deve ser reduzida na mesma proporção.

Os resultados de simulação retratados em (KOZEK, ROSKA e CHUA, 1993) com três exemplos de operações típicas em processamento de imagens permitiram uma análise comparativa das possíveis opções para cada etapa do algoritmo. A conclusão obtida atribui o melhor desempenho, no geral, à utilização da codificação aprimorada, da avaliação pelo janelamento e do elitismo. Quanto à recombinação, o cruzamento em dois pontos mostrou uma melhor eficiência em casos que não envolvem muitos coeficientes, perdendo a primazia para o cruzamento aleatório à medida que se aumenta o tamanho do cromossomo.

2.3 FILTRAGEM DE IMAGENS

A filtragem corresponde a uma das operações mais frequentes no processamento de imagens e sua utilização pode estar presente em diversas etapas da computação, cumprindo as mais diversas finalidades. Exemplos incluem o realce de determinadas características da imagem, como bordas ou objetos, e a suavização de imagens.

Muitos dos conceitos utilizados na filtragem, explanados em (GONZALEZ e WOODS, 2007), originam-se do campo de processamento de sinais, tirando-se proveito inclusive dos formalismos matemáticos utilizados para este fim, com a ressalva de que no caso do tratamento de imagens as grandezas envolvidas são representadas por sinais variantes no espaço, ao invés de possuírem uma dependência no tempo.

Dessa forma, a operação de filtragem aplicada a imagens pode ser abordada de duas maneiras: no domínio do espaço e no domínio da frequência, ambas proporcionando diferentes métodos para o projeto de filtros. Apesar destas alternativas possuírem uma correlação matemática, a escolha mais adequada pode depender da aplicação desejada ou dos recursos disponíveis.

2.3.1 Domínio Espacial

No domínio espacial, um filtro é definido por um conjunto de coeficientes formando uma vizinhança e aplicados em uma operação pré-determinada. Para cada pixel da imagem, esta operação será feita no seu entorno, e o seu resultado corresponderá ao valor do pixel com as mesmas coordenadas na imagem de saída. O filtro espacial pode ser linear ou não-linear, de acordo com sua operação.

Seja o diagrama da FIGURA 2.19 a representação de uma máscara de coeficientes de um filtro linear de tamanho 3 por 3. Nesta seção, assim como no restante do trabalho, são consideradas imagens monocromáticas. Contudo, este desenvolvimento pode ser aproveitado para a aplicação envolvendo imagem em cores. Além disso, nesta análise o valor 0 corresponde a um pixel preto e o valor máximo do pixel, que em aplicações digitais depende no número de bits usado para representação, está associado à cor branca. Para cada pixel, é realizado o produto de cada coeficiente w da máscara pelo valor do pixel vizinho correspondente, obedecendo à sua posição relativa ao centro; o pixel central é multiplicado pelo coeficiente central. Os resultados são então somados, produzindo o valor do pixel da imagem filtrada. A equação (2.26) representa este procedimento, com $f(x,y)$ e $\mathrm{g}(x,y)$ correspondendo aos valores dos pixels das coordenadas (x,y) para a

imagem antes e depois da filtragem, respectivamente; a orientação segue como mostrado na FIGURA 2.20.

$w_{-1,-1}$	$w_{-1,0}$	$w_{-1,1}$
$w_{0,-1}$	$w_{0,0}$	$w_{0,1}$
$w_{1,-1}$	$w_{1,0}$	$w_{1,1}$

FIGURA 2.19. Máscara de um filtro 3x3.

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \cdots$$

$$+ w(0,0)f(x,y) + \cdots + w(1,0)f(x+1,y) + w(1,1)f(x+1,y+1) \qquad (2.26)$$

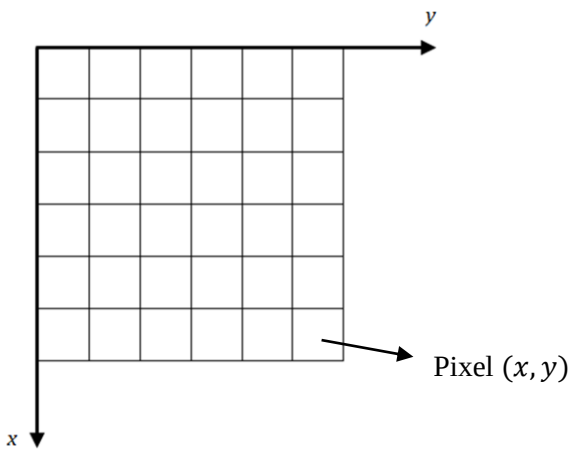


FIGURA 2.20. Sistema de coordenadas para os pixels.

De um modo geral, para uma imagem de tamanho M x N, sendo M e N ímpares, e um filtro de tamanho m x n, a equação (2.27) representa a operação de filtragem linear. Os índices a e b são tais que $m = 2a + 1$ e $n = 2b + 1$; s e t são variáveis auxiliares.

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \qquad (2.27)$$

O procedimento anterior está intimamente relacionado a dois importantes conceitos: correlação e convolução. O primeiro equivale à operação descrita na equação (2.27) e tem papel de destaque nas aplicações que envolvem busca de correspondência entre imagens. Já a convolução difere apenas na utilização da máscara, que sofre uma rotação de 180 graus antes da operação, e serve como base para a formulação da teoria que relaciona matematicamente o domínio espacial e o domínio da frequência. As equações (2.28) e (2.29) descrevem, respectivamente, as operações de correlação e de convolução, incluindo a simbologia usual. Pode-se mostrar que, nos casos de filtros cuja máscara possui simetria radial, ambas operações produzem o mesmo resultado.

$$w(x,y) \circ f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x+s, y+t) \tag{2.28}$$

$$w(x,y) * f(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t) f(x-s, y-t) \tag{2.29}$$

2.3.2 Domínio da Frequência

A abordagem no domínio da frequência é fundamentada a partir do uso da transformada de Fourier para duas dimensões sobre o domínio espacial. Para a adequação à forma de representação das imagens, dividas em pixels, utiliza-se a sua versão discreta. A definição do par com a transformação direta (DFT-2D: *Two Dimensional Discrete Fourier Transform*) e a inversa (IDFT-2D: *Two Dimensional Inverse Discrete Fourier Transform*) é mostrada nas equações (2.30):

$$\begin{cases} \tilde{F}(\omega, \psi) = \Im[f(x,y)] = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) e^{-j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} & (2.30.a) \\ f(x,y) = \Im^{-1}[\tilde{F}(\omega, \psi)] = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \tilde{F}(\omega, \psi) e^{j2\pi\left(\frac{ux}{M} + \frac{vy}{N}\right)} & (2.30.b) \end{cases}$$

onde u é a componente de frequência horizontal, v é a componente de frequência vertical (ambas dadas em ciclos/pixel), $\Im$ e $\Im^{-1}$ representam respectivamente as operações de transformação direta e inversa e $\tilde{F}(\omega, \psi)$ é a transformada da imagem $f(x,y)$. Ambas transformações geram sequências numéricas com domínio infinito e periódicas nas

direções horizontal e vertical, com períodos iguais a M e N, respectivamente. Para fins de filtragem, pode-se considerar apenas um período das funções.

A filtragem no domínio da frequência consiste em modificar a DFT da imagem utilizando uma função de transferência que representa o filtro, seguida da IDFT para obtenção da imagem resultante. Este processo é representado matematicamente pela equação (2.31), onde $\tilde{F}(\omega, \psi)$ é um período da DFT da imagem, $\tilde{H}(\omega, \psi)$ é a função de transferência do filtro (ambas funções possuem dimensões M por N e $g(x, y)$ é a imagem de saída.

$$g(x, y) = \Im^{-1}\left[\tilde{H}(\omega, \psi)\tilde{F}(\omega, \psi)\right] \tag{2.31}$$

Neste processo, a DFT da imagem terá seus elementos multiplicados pela função do filtro e, com base nesse produto, filtros podem ser projetados para se trabalhar nas regiões de frequência que se deseja modificar. Por questão de simplicidade, costuma-se utilizar $\tilde{H}(u, v)$ simétrica em relação ao centro. Para isso é necessária também a centralização dos elementos de $\tilde{F}(\omega, \psi)$, a qual pode ser facilmente obtida multiplicando $f(x, y)$ por $(-1)^{x+y}$ antes de aplicar a DFT. Este procedimento, exemplificado na FIGURA 2.21 para uma transformada com $M = N = 50$, desloca os elementos que correspondem às menores frequências para o centro da matriz e, a partir deste ponto, a frequência crescerá radialmente, atingindo os maiores valores nas suas extremidades.

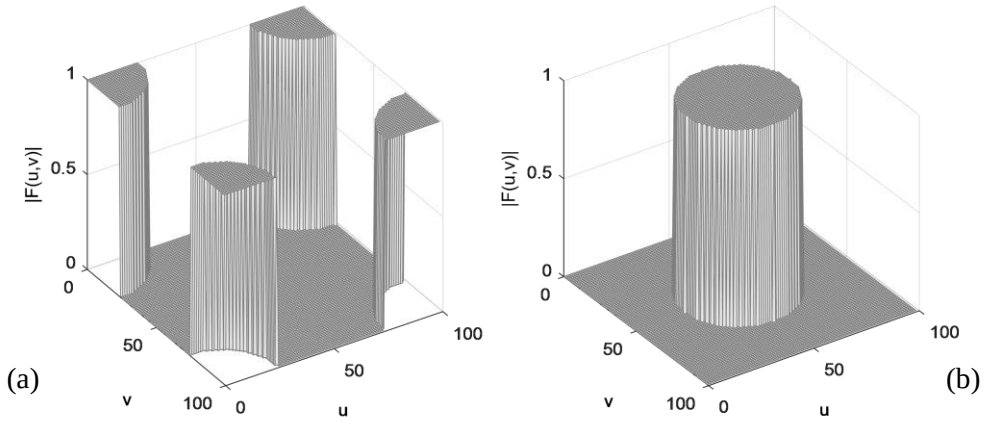


FIGURA 2.21. Exemplo do processo de centralização: (a) Forma comum de $\tilde{F}(u, v)$; (b) Forma após a centralização . ω e ψ são dados em ciclos/pixel. Obtidas por meio do *software* Matlab®.

Um detalhe que deve ser levado em conta nesta metodologia é que para evitar efeitos indesejados na filtragem devidos à periodicidade da DFT da imagem, faz-se necessário o preenchimento de zeros concatenados nas bordas de $f(x, y)$ (*padding*), que correspondem a pixels pretos, de forma que a imagem fique com um tamanho $P \times Q$, de acordo com as inequações em (2.32). A função do filtro neste caso também deve respeitar estas dimensões. Tipicamente é estipulado $P = 2M$ e $Q = 2N$. A FIGURA 2.22 mostra um exemplo deste procedimento.

$$\begin{cases} P \geq 2M - 1 \\ Q \geq 2N - 1 \end{cases} \tag{2.32}$$

25	12	23	15
24	57	85	45
55	45	21	18
36	21	01	08

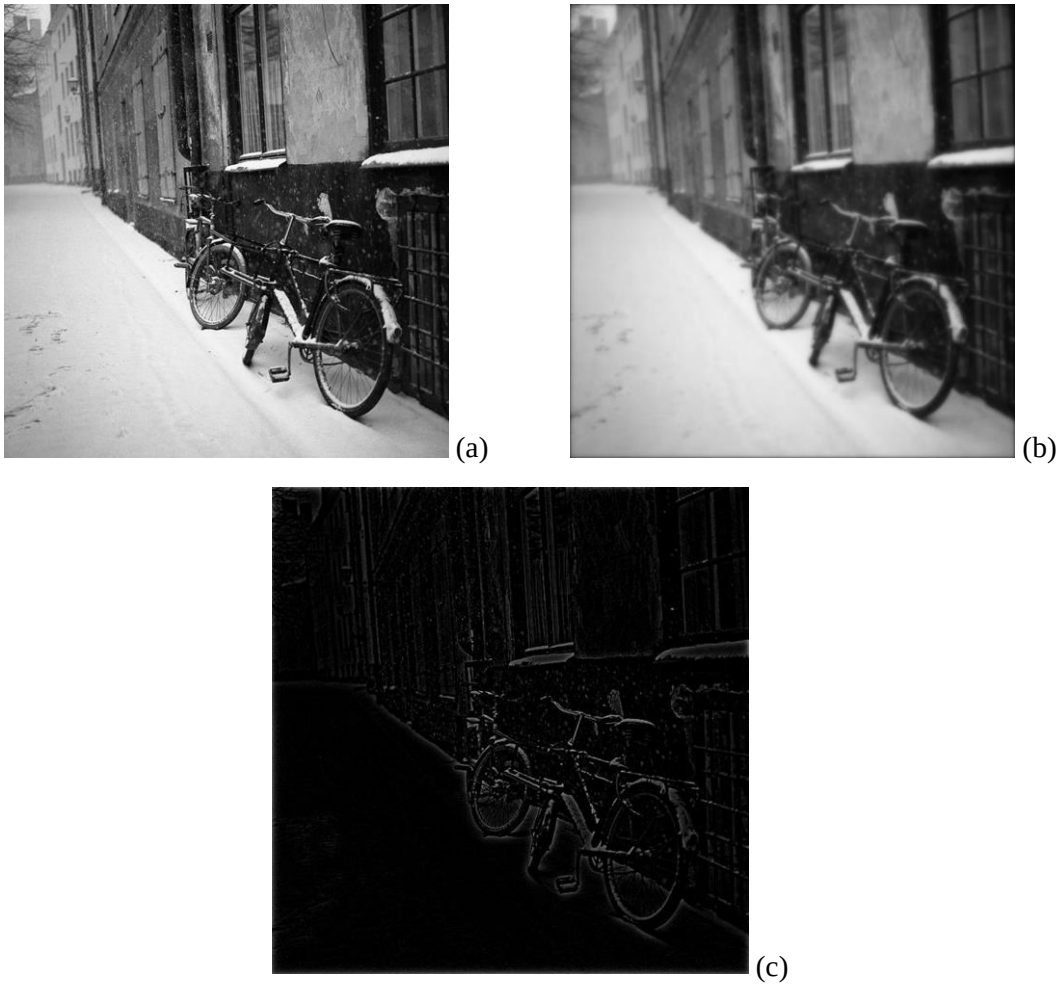
25	12	23	15	0	0	0	0
24	57	85	45	0	0	0	0
55	45	21	18	0	0	0	0
36	21	01	08	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

FIGURA 2.22. Preenchimento de zeros nas bordas uma imagem. A imagem à esquerda, com seus pixels representados numericamente, é ampliada com pixels nulos, formando a imagem à direita.

Filtros básicos podem ser projetados a partir de funções circulares e concêntricas. As baixas frequências correspondem às transições gradativas das tonalidades. Logo, um filtro passa-baixas pode ser usado para a suavização de imagens. Por sua vez, as altas frequências estão associadas às transições abruptas, como as bordas de objetos. Consequentemente, um filtro passa-altas realçará estes traços, sendo útil em aplicações que envolvam a detecção de bordas. Eliminar apenas a frequência DC, no centro da transformada, anula o valor da intensidade média da imagem. Exemplos destes três processos são mostrados nas FIGURAS 2.23 e 2.24. Filtros mais complexos, como passa-faixa e rejeita-faixa, podem ser obtidos a partir de combinações dos casos anteriores.

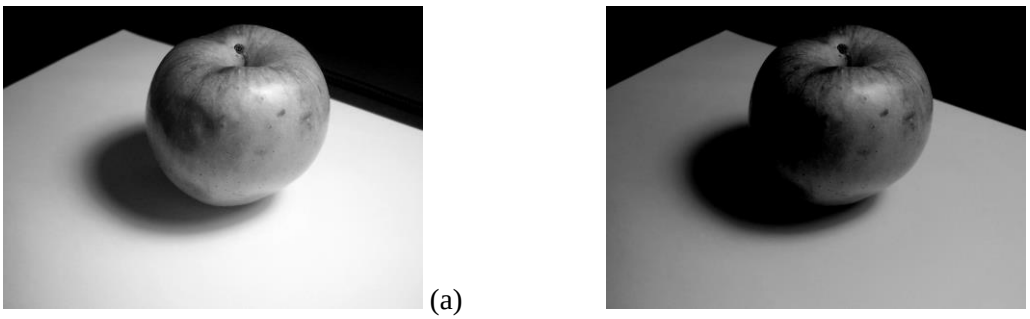
O uso de uma função $\widetilde{H}(u, v)$ baseado na forma de um filtro ideal, apesar de ser possível, possui limitações que impedem um desempenho perfeito, decorrente da

transição abrupta entre as bandas de passagem e de rejeição. Uma forma de reduzir este problema é o uso de aproximações que possuam uma faixa de transição contínua, como o caso dos filtros de Butterworth e gaussiano, descritos a seguir.



(a) (b) (c)

FIGURA 2.23. Exemplo de imagens filtradas: (a) Imagem original; (b) Imagem processada por um filtro passa-baixas; (c) Imagem processada por um filtro passa altas.



(a) (b)

FIGURA 2.24. Exemplo de filtragem para eliminar o termo DC: (a) Imagem original; (b) Imagem filtrada.

2.3.2.1 Filtro passa-baixas de Butterworth

A função de transferência deste tipo de filtro é descrita na expressão (2.33). A variável n é a ordem do filtro, que determina a inclinação da banda de transição; filtros de maior ordem terão um decaimento mais rápido, gerando uma zona de transição mais curta e aproximando-se do caso ideal. A função $D(\omega, \psi)$ é a distância de um ponto (ω, ψ) para o centro do filtro, obtida a partir da relação (2.34). Já D_0 é a frequência de corte do filtro, definida como o raio do círculo que contém os pontos onde o valor de $\widetilde{H}_{PB}$ corresponde à metade de seu valor máximo (equivalente a 1 para os filtros abordados neste trabalho). A FIGURA 2.25 traz uma representação gráfica da magnitude da função, bem como o seu perfil radial para diferentes ordens.

$$\widetilde{H}_{PB}(\omega, \psi) = \frac{1}{1 + \left(\frac{D(\omega, \psi)}{D_0}\right)^{2n}} \tag{2.33}$$

$$D(\omega, \psi) = \sqrt{\left[\left(u - \frac{P}{2}\right)^2 + \left(v - \frac{Q}{2}\right)^2\right]} \tag{2.34}$$

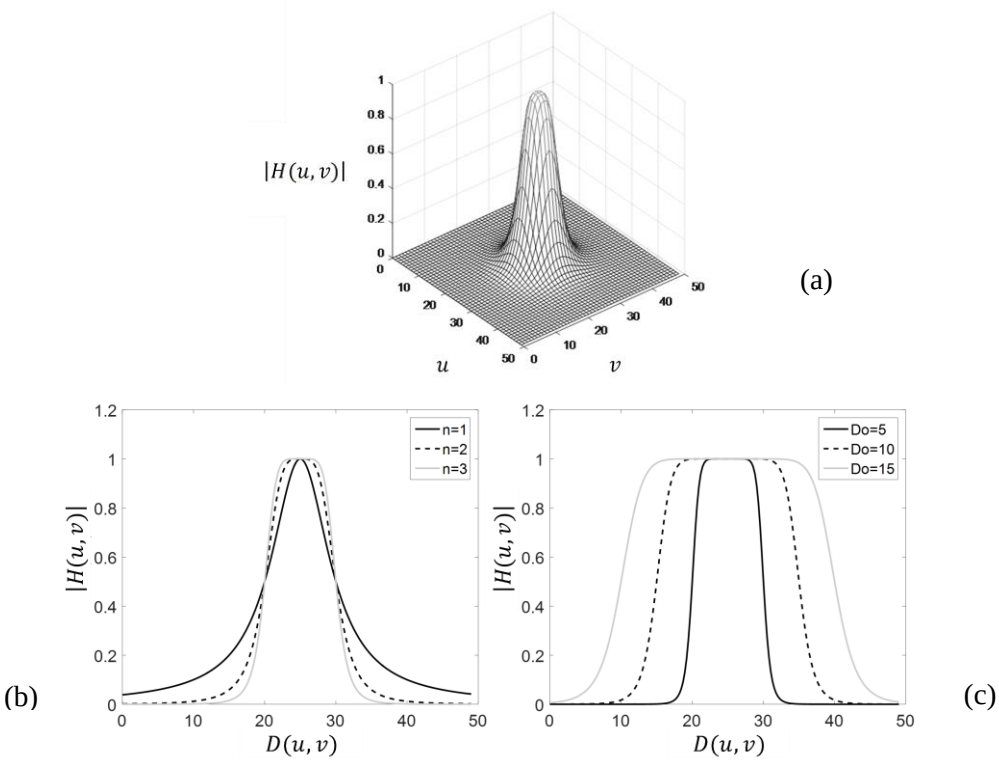


FIGURA 2.25. Representação gráfica do filtro passa-baixas de Butterworth: (a) Função de transferência, com $Do = 10$, $n = 2$ e $M = N = 50$; (b) Perfis radiais da função para diferentes ordens n; (c) Perfis radiais da função para diferentes frequências de corte D_0. As variáveis ω e ψ são dadas em ciclos/pixel. Obtidos por meio do *software* Matlab®.

2.3.2.2 Filtro passa-altas de Butterworth

Sua função pode ser obtida subtraindo-se $\widetilde{H}_{PB}$ de 1, conforme mostrado na expressão (2.35), e portanto possui parâmetros idênticos à versão passa-baixas. A FIGURA 2.26 descreve seu formato.

$$\widetilde{H}_{PA}(\omega,\psi) = 1 - \widetilde{H}_{PB} = -\frac{1}{1+\left(\frac{D_0}{D(\omega,\psi)}\right)^{2n}} \tag{2.35}$$

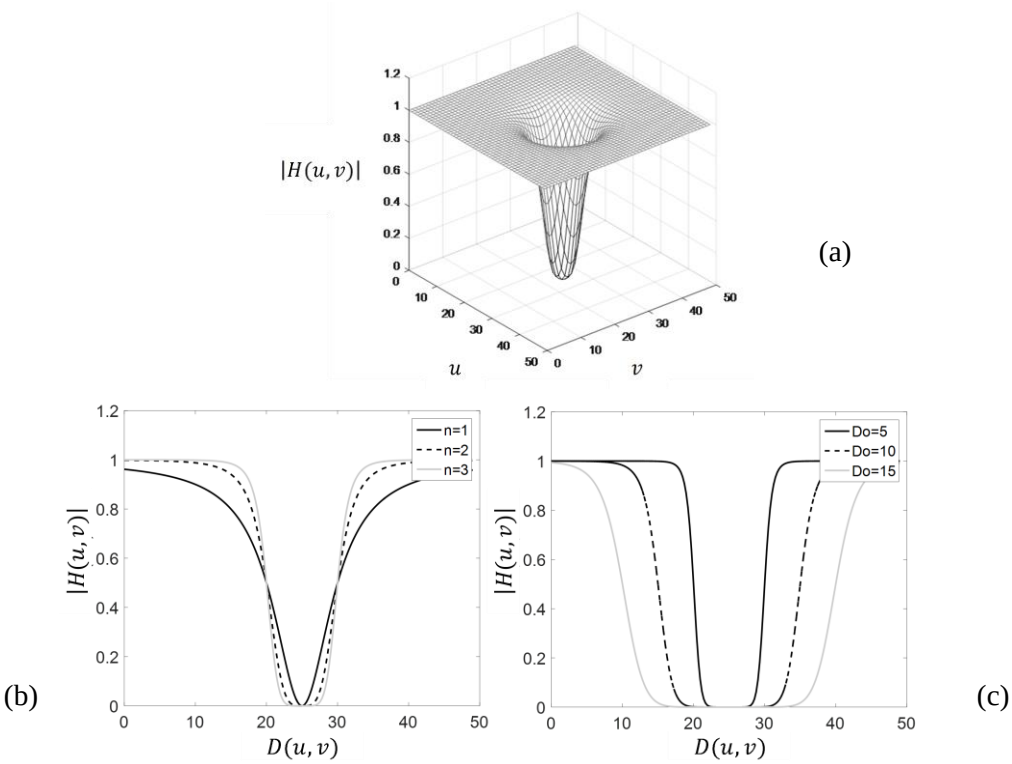


FIGURA 2.26. Representação gráfica do filtro passa-baixas de Butterworth: (a) Função de transferência, com $Do = 10$, $n = 2$ e $M = N = 50$; (b) Perfis radiais da função para diferentes ordens n; (c) Perfis radiais da função para diferentes frequências de corte D_0. As variáveis ω e ψ são dadas em ciclos/pixel. Obtidos por meio do *software* Matlab®.

2.3.3 Teorema da Convolução 2D

O clássico teorema da convolução consiste em um elemento crucial do campo de processamento de sinais e sua extensão para uma abordagem em duas dimensões permite a formação de uma ligação matemática entre ambos os domínios tratados anteriormente. De forma sucinta, a mesma estabelece que:

$$\begin{cases} f(x,y) * h(x,y) = \tilde{F}(\omega,\psi)\tilde{H}(\omega,\psi) & (2.36.a) \\ f(x,y)h(x,y) = \tilde{F}(\omega,\psi) * \tilde{H}(\omega,\psi) & (2.36.b) \end{cases}$$

indicando que a convolução entre duas funções espaciais $f(x,y)$ e $h(x,y)$ corresponde a uma multiplicação entre os elementos de suas DFT, $\tilde{F}(\omega,\psi)$ e $\tilde{H}(\omega,\psi)$, ou vice-versa. Uma condição importante para a garantia da validade das relações (2.36) é que f e h passem pelo processo de preenchimento de zeros, a fim de que a periodicidade da DFT não interfira no resultado do lado direito das equações.

Deste teorema infere-se, portanto, uma correspondência direta entre as duas metodologias de projeto de filtros mencionadas nos itens anteriores, auxiliando também na percepção da proximidade entre esta classe de operação e a dinâmica da CNN.

2.4 ANÁLISE DA RESPOSTA DA CNN NO DOMÍNIO DA FREQUÊNCIA

As ferramentas matemáticas introduzidas na seção anterior também são úteis para uma análise da resposta de uma CNN sob a ótica da frequência. A combinação da transformada de Fourier com o teorema da convolução permitem reescrever a dinâmica da rede de uma forma mais direta, para certas condições. Este desenvolvimento, descrito em (CHUA e ROSKA, 2002), ajuda a elucidar algumas características deste tipo de sistema que o tornam conveniente para determinadas aplicações, em especial a filtragem de imagens, e podem contribuir para facilitar o seu emprego.

Algumas condições são previamente estabelecidas: a simetria radial das matrizes de operadores sinápticos, A e B, e a manutenção dos níveis de estado das células na faixa em que a função de saída é linear $[-1,1]$. Partindo da equação (2.1) e reescrevendo os somatórios, tem-se que:

$$\dot{x_{i,j}} = -x_{i,j} + \left[\sum_{k=-R}^{R}\sum_{l=-R}^{R} a_{k,l}y_{(i+k),(j+l)}\right] + \left[\sum_{k=-R}^{R}\sum_{l=-R}^{R} b_{k,l}u_{(i+k),(j+l)}\right] + z_{i,j} \qquad (2.37)$$

Desta forma, considerando a simetria de A e B e o fato da relação (2.2) se reduzir a $y_{i,j} = x_{i,j}$, chega-se a:

$$\dot{x_{ij}} = \left[\sum_{k=-R}^{R}\sum_{l=-R}^{R} a'_{-k,-l}y_{(i+k),(j+l)}\right] + \left[\sum_{k=-R}^{R}\sum_{l=-R}^{R} b_{-k,-l}u_{(i+k),(j+l)}\right] + z_{i,j} \qquad (2.38)$$

onde $a'_{k,l}$ é dado por:

$$a'_{k,l} = \begin{cases} a_{k,l} - 1, & k = l = 0 \\ a_{k,l}, & caso\ contrário \end{cases} \tag{2.39}$$

O conjunto formado pela equação (2.38) referente a cada célula da rede forma um sistema de equações diferenciais acopladas. Associando os somatórios à operação de convolução, aplicando a DFT e o teorema da convolução (2.36.a), aos seus termos, tem-se que:

$$\frac{d\tilde{X}_t(\omega,\psi)}{dt} = \tilde{A}'(\omega,\psi)\tilde{X}_t(\omega,\psi) + \tilde{B}(\omega,\psi)\tilde{U}(\omega,\psi) + \tilde{Z}(\omega,\psi) \tag{2.40}$$

o que representa um sistema de equações desacopladas, facilitando a sua resolução. Vale ressaltar que as operações aplicadas anteriormente atêm-se ao domínio espacial em duas dimensões, não removendo o caráter temporal do equacionamento. Deste modo, a solução corresponde, no domínio da frequência, a:

$$\tilde{X}_t(\omega,\psi) = e^{\tilde{A}'(\omega,\psi)t}\tilde{X}_0(\omega,\psi) + \frac{1}{\tilde{A}'(\omega,\psi)}\left(e^{\tilde{A}'(\omega,\psi)t} - 1\right)\left(\tilde{B}(\omega,\psi)\tilde{U}(\omega,\psi) + \tilde{Z}(\omega,\psi)\right) \tag{2.41}$$

onde $\tilde{X}_0$ é a DFT do estado inicial das células para $t = 0$. A equação (2.41) pode então ser aplicada para obtenção da resposta final da CNN em alguns casos especiais. Se $Re\{\tilde{A}'(\omega,\psi)\} < 0$, tem-se:

$$\tilde{X}_\infty(\omega,\psi) = \lim_{t\to\infty} \tilde{X}_t(\omega,\psi) = -\frac{1}{\tilde{A}'(\omega,\psi)}\left(\tilde{B}(\omega,\psi)\tilde{U}(\omega,\psi) + \tilde{Z}(\omega,\psi)\right) \tag{2.42}$$

Para a condição específica em que $\tilde{A}'(\omega,\psi) = -1$, que implica em uma matriz A com coeficientes todos nulos, a resposta é dada pela equação (2.43):

$$\tilde{X}_\infty(\omega,\psi) = \tilde{B}(\omega,\psi)\tilde{U}(\omega,\psi) + \tilde{Z}(\omega,\psi) \tag{2.43}$$

Estas duas últimas equações permitem a formação de associações bem interessantes entre a dinâmica de uma CNN e as operações de filtragem de imagens. De fato, descartando o uso do coeficiente de limiar ($\tilde{Z}(\omega,\psi) = 0$), a relação (2.43) torna-se:

$$\tilde{X}_\infty(\omega,\psi) = \tilde{B}(\omega,\psi)\tilde{U}(\omega,\psi) \tag{2.44}$$

o que corresponde à aplicação de uma imagem fornecida na entrada da rede a um filtro FIR cuja função de transferência equivale a $\tilde{B}(\omega,\psi)$. Por sua vez, desprezando-se novamente $\tilde{Z}(\omega,\psi)$, este mesmo tratamento efetuado em (2.42) resulta em:

$$\tilde{X}_{\infty}(\omega,\psi) = -\frac{\tilde{B}(\omega,\psi)}{\tilde{A}'(\omega,\psi)}\tilde{U}(\omega,\psi) \tag{2.45}$$

Nesta situação, a função de transferência do filtro equivalente é composta pela razão entre $\tilde{B}$ e $\tilde{A}$, possibilitando, a princípio, contornar a limitação dimensional das matrizes sinápticas, no domínio do espaço, e por conseguinte indicando uma capacidade teórica da CNN para reproduzir operações de filtragem IIR. Neste contexto, entretanto, a formulação de uma abordagem analítica para definição dos coeficientes da rede vinculados ao filtro projetado esbarra na dificuldade de formar uma composição das matrizes sinápticas que satisfaça simultaneamente, no domínio da frequência, a função de transferência desejada e as condições consideradas no desenvolvimento que culmina na expressão (2.45).

Destaca-se também aqui o fato de que o emprego de células do tipo FSR para a implementação da CNN não invalida as constatações presentes nesta seção, sendo a única distinção relevante a presença de $\tilde{A}$, a DFT da própria matriz A, em substituição a $\tilde{A}'$.

2.5 UM CIRCUITO ANALÓGICO DE CNN DO TIPO FSR EM TECNOLOGIA CMOS

O desempenho de uma CNN depende em grande parte do funcionamento das sinapses. Portanto, o projeto de um circuito eficiente para esta função é imprescindível para a obtenção de uma arquitetura que corresponda às necessidades das aplicações tratadas neste trabalho. Tal desenvolvimento, documentado em (SANTANA, 2013), produziu um circuito que emprega o reaproveitamento de blocos para redução de tamanho e de consumo, dentre outras vantagens.

Nesta abordagem, aplicada para a construção de uma rede do tipo FSR, a sinapse é realizada a partir de multiplicadores em modo tensão-corrente, com entradas em corrente e tensão e saída em corrente, cujo núcleo é formado a partir da estrutura representada na FIGURA 2.27. A operação é feita a partir do espelhamento da transcondutância de fonte entre os transistores M_1 e M_{3A}, considerando ambos operando no início da região triodo. Os demais transistores operam na região de saturação. Pode-se

demonstrar que o valor da corrente na saída i_{outA}, em termos dos sinais de entrada v_{in} e i_{in}, é dado pela expressão (2.46):

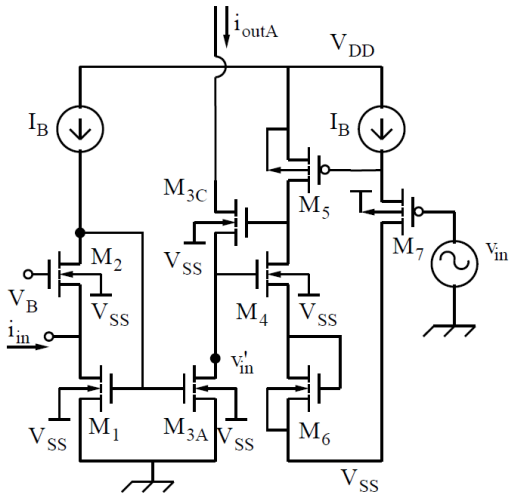


FIGURA 2.27. Núcleo do multiplicador. Extraído de (SANTANA, 2013).

$$i_{outA} = \frac{i_{in}+I_B}{V_{DS1}}(V_{IDC} + kv_{in}) \qquad (2.46)$$

onde V_{DS1} é a tensão dreno-fonte de M_1, I_B é uma corrente de polarização, V_{IDC} é o deslocamento de nível total desde o terminal de porta do transistor M_7 até o terminal de porta de M_4 e k é um fator gerado pelo efeito de corpo.

Os níveis de polarização do circuito produzem termos indesejados no produto, que são cancelados a partir da combinação de produtos e subtrações envolvendo os sinais do lado direito da relação em (2.46).

Com base neste núcleo, em (SANTANA, 2013) foi concebida uma arquitetura para uma rede neuronal celular analógica do tipo FSR, cujas células são constituídas pelos circuitos da FIGURA 2.28. Nesta rede utiliza-se para as sinapses uma versão do multiplicador com operação em quatro quadrantes, composto por quatro núcleos que compartilham parte de sua estrutura e por um subtrator para efetuar o cancelamento citado anteriormente.

Os coeficientes da rede são introduzidos em cada sinapse a partir do sinal de corrente i_{in}, no bloco gerador de peso. Por sua vez, os sinais de entrada e os representativos dos estados das células assumem, em cada sinapse, a forma da tensão v_{ish}. Os multiplicadores processam essas grandezas e produzem como resultado as correntes designadas i_{out}, em cada sinapse, sendo inseridas no nó X. Portanto, a confluência destas

correntes corresponde ao somatório dos termos referentes aos coeficientes das matrizes *A* e *B* conforme (2.3), além do limiar, que também é implementado a partir de um bloco de sinapse adicional. Neste mesmo nó é conectado também um grampeador, idêntico ao utilizado em (HEGT, LEENAERTS e WILMANS, 1998), para produzir $gp(x)$, como descrito em (2.4). Adicionalmente, a integração é facilitada ao ser realizada por meio da capacitância total no nó X, dispensando a necessidade de um circuito específico. Finalmente, a saída da célula corresponde ao sinal de tensão v_o, após passar por um deslocamento de nível.

Uma CNN em tecnologia CMOS de comprimento mínimo igual a 0,13 µm, com base nesta arquitetura, também foi concebida em (SANTANA, FREIRE e CUNHA, 2012), sendo posteriormente fabricada. Uma versão de dimensões 10x10 pré-leiaute foi utilizada para as simulações relatadas em (SANTANA, 2013) e (ANDRADE, 2015a), cujos resultados apresentam, respectivamente, casos de reprodução fiel de operações bipolares e a realização da filtragem de imagens com erros pequenos.

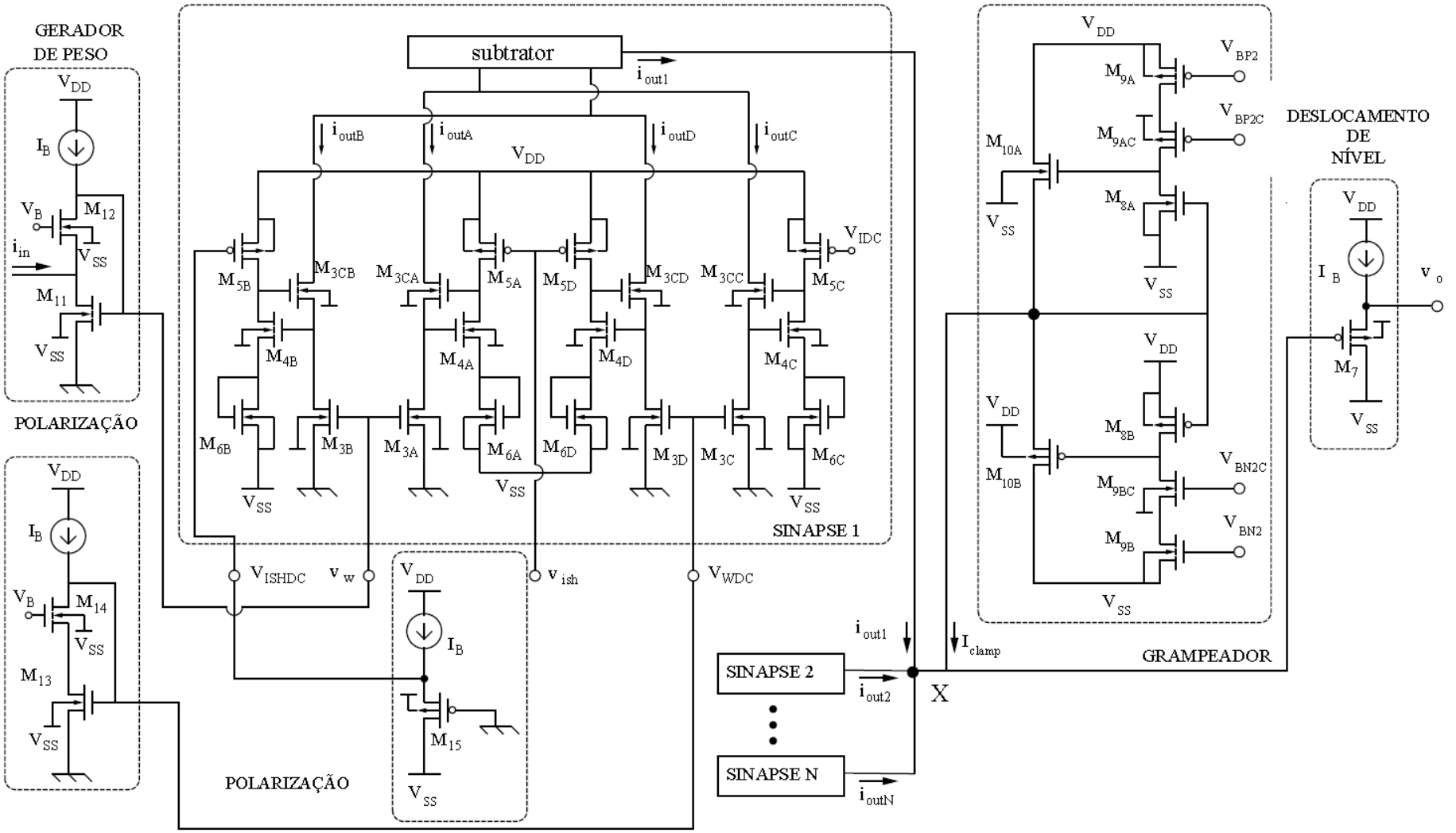


FIGURA 2.28. Circuito da célula da CNN. Extraído de (SANTANA, 2013).

3 TREINAMENTO DE CNN

Neste capítulo serão descritas as inovações e adaptações introduzidas nos algoritmos escolhidos para realizar o treinamento da CNN, bem como os desafios e dificuldades que as motivaram, especialmente os relacionados à filtragem de imagens e à operação da CNN com duas camadas. Uma análise comparativa entre os métodos de treinamento abordados é apresentada ao final do capítulo.

3.1 IMPLEMENTAÇÃO

3.1.1 Algoritmo do Centro de Massa

3.1.1.1 CNN com uma camada

O CMA foi implementado anteriormente em (ANDRADE, 2015a) via código executável no ambiente do *software* Matlab®, a princípio sem adaptações. O treinamento foi realizado em um modelo ideal da CNN FSR, cuja simulação foi realizada com o uso da técnica de integração numérica a partir da fórmula de Euler. Dessa forma o estado da rede é variado a cada intervalo de tempo Δt de acordo com a equação (3.1):

$$x(t + \Delta t) \cong x(t) + \Delta t \dot{x}(t) \tag{3.1}$$

onde $\dot{x}(t)$ é dada por (2.1). A relação (3.1) é qualitativamente correta e precisa o suficiente se utilizarmos um passo temporal Δt de baixa ordem de grandeza, sendo adotado neste trabalho o valor de $0{,}1\ s$, representando de acordo com (CHUA e ROSKA, 2002), um compromisso adequado entre precisão e robustez e portanto podendo ser aplicável para os tipos de operações tratados neste trabalho. Além disso, a representação de ponto flutuante de 64 bits utilizada para os sinais da rede forneceu uma precisão adequada para eliminar a influência da discretização no funcionamento da CNN.

Os testes iniciais do CMA envolveram funções bipolares bem simples e a implementação funcionou de forma satisfatória, confirmando os resultados com convergência total (sem erros) mostrados em (MIRZAI, CHENG e MOSCHYTZ, 1998). Entretanto, nos experimentos visando a obtenção dos coeficientes da rede para casos com dinâmica mais elaborada, como as operações que envolvem propagação ou a filtragem de imagens, algumas dificuldades foram observadas quanto à velocidade e à convergência. Diante deste cenário, tais limitações foram tratadas a partir da implementação de modificações no CMA.

Um ponto de destaque do processamento com filtros de imagens que requer atenção é o fato de o mesmo geralmente não envolver funções bipolares, ou seja, a saída de cada célula pode se estabilizar em qualquer nível entre -1 e 1, correspondendo a tons de cinza mapeados de forma linear, como representado pela FIGURA 3.1. Essa faixa contínua de valores tende a restringir o conjunto de possíveis soluções para a maioria dessas funções, uma vez que uma elevada precisão da resposta pode requerer combinações de coeficientes com valores exatos. Neste contexto, as limitações inerentes à aplicação dos métodos numéricos impactam de maneira mais significativa em funções não-bipolares, impedindo que se anule totalmente o erro e exigindo o estabelecimento de uma tolerância. Ademais, como está eventualmente acompanhada de uma maior quantidade de coeficientes ajustáveis, a aplicação de métodos numéricos demanda um maior esforço no aprendizado.

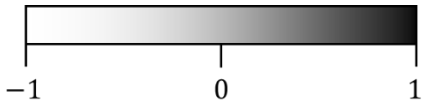


FIGURA 3.1. Representação do valor dos pixels para funções em tons de cinza.

Para melhorar o desempenho do treinamento, ainda em (ANDRADE, 2015a), foram analisados alguns aspectos do método que prejudicavam o seu funcionamento, culminando na elaboração de algumas modificações no algoritmo original e gerando uma versão mais robusta.

Um problema crítico observado consiste na baixa velocidade das variações dos coeficientes, exigindo do CMA uma exagerada quantidade de iterações (na ordem de 100 vezes mais iterações), se comparada à quantidade de iterações requeridas pelas funções binárias. O desempenho nesse aspecto foi melhorado, ao ser introduzida uma opção para a taxa de aprendizado ser variável a partir de um valor inicial, seguindo um movimento linear e decrescente. Esta abordagem permite o uso de taxas altas nos passos iniciais da otimização, acelerando esta parte do processo, enquanto mantém simultaneamente um aumento gradual da precisão de busca a cada passo.

Mesmo com a aplicação desta medida, ainda há casos em que a busca pela solução fica estagnada, com o algoritmo dando voltas na mesma região enquanto o erro se mantém acima da tolerância, o que pode ocorrer em mínimos locais. Para mitigação desta deficiência, além do critério tradicional que verifica o erro de cada célula com uma tolerância determinada, incluiu-se uma segunda condição para interrupção das iterações

do treinamento. Esta condição verifica se há uma trajetória cíclica em cada uma das matrizes de coeficientes e no limiar, ao observar possíveis repetições em seus valores em um número fixo de iterações anteriores mais recentes.

Adicionalmente, foi inserida a possibilidade de manter a componente simétrica da matriz A com um padrão fixo durante o treinamento, de forma análoga ao já sugerido anteriormente para a matriz B e ilustrado nas equações (2.12) e (2.13), o que é conveniente para acelerar o processo para muitas funções, incluindo os filtros espaciais, onde tal simetria já é previamente estabelecida. E, por último, a programação permite a aplicação automática de mais de um conjunto de imagens, sendo que o treinamento é executado para cada conjunto e o resultado é utilizado como condição inicial para o seguinte. Isto permite um ajuste mais amplo dos coeficientes, ao serem utilizadas imagens que realcem diferentes particularidades da função.

Tais ajustes incrementaram o desempenho do algoritmo, permitindo sua utilização em aplicações de filtragem de imagens, auxiliada por uma ferramenta para treinamento de CNN com interface gráfica desenvolvida conforme mostrado em (ANDRADE, 2015a), onde foram obtidas respostas adequadas para o processamento nas simulações do circuito que foi objeto de avaliação.

3.1.1.2 Abordagem para o treinamento da 2L-CNN

O CMA foi originalmente desenvolvido tendo em vista o uso em CNN simples, com uma camada. Todavia, sua estrutura baseada nas técnicas de retropropagação aplicadas em RNA, que geralmente possuem várias camadas, a tornam propícia para ser estendida de forma análoga às aplicações com CNN de múltiplas camadas. Assim, o algoritmo foi adaptado e sua capacidade foi ampliada para tratar da 2L-CNN.

Apesar de ser um processo aparentemente intuitivo, alguns aspectos merecem consideração. O primeiro deles é a inclusão das matrizes de coeficientes C_n, existentes nas equações (2.6), o que requer a definição de como serão realizadas suas atualizações. A princípio, pelo fato de representarem pesos sinápticos vinculados a sinais de saída, de forma análoga aos coeficientes de A, seria intuitivo moldar a variação em cada iteração conforme (2.11.a) levando em conta a decomposição retratada em (2.15.c). Entretanto, o mecanismo envolvido na componente antissimétrica para mover o centro de massa perde significado em virtude das sinapses referentes à C receberem os sinais da outra camada.

Consequentemente, se tornou mais conveniente definir sua atualização com base no que é realizado para a matriz B, definida pela equação (2.11.b), para tal propósito.

Outro ponto importante, que merece destaque, é o processamento do erro. Em funções para as quais se tem os valores esperados das duas camadas, como as descritas em (YANG, NISHIO e USHIDA, 2003), o que ocorre geralmente com os casos em que há o emprego de duas funções simples em sequência ou operações simultâneas nos dois níveis, basta utilizar uma matriz de erros para cada camada sem nenhuma alteração nos cálculos. Contudo, em outras situações, como aquelas envolvendo filtragem de imagens, pode ser conveniente reproduzir o arranjo empregado em outras arquiteturas de RNA, como as redes convolucionais, e treinar a CNN tendo como referência apenas a saída da última camada, ignorando para fins de avaliação de erros a resposta das demais células. Um exemplo desta abordagem está ilustrado na FIGURA 3.2. Tomando como base as relações (2.11), este fato gera uma lacuna nos cálculos das variações dos coeficientes da primeira camada, exigindo um tratamento adicional. A solução passou pelo aproveitamento do fato de o CMA se inspirar na técnica de retropropagação, cujo princípio consiste na transferência do erro, no sentido inverso do fluxo de processamento convencional (da última camada para a primeira), ponderado pelos pesos referentes às conexões entre as células percorridas de forma cumulativa. Ao aplicar esse conceito à 2L-CNN, obtêm-se a partir das expressões (2.11) as relações (3.2) a seguir:

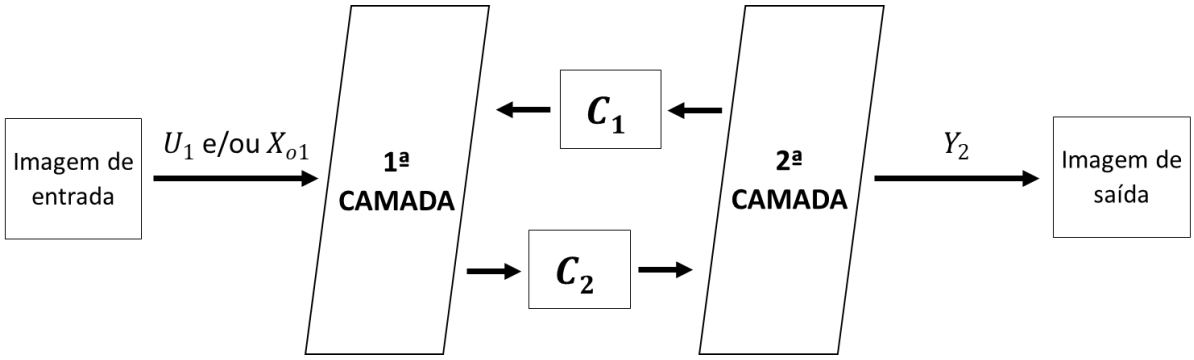


FIGURA 3.2. Exemplo da aplicação dos sinais no treinamento da 2L-CNN para filtragem de imagens.

$$\Delta a_{l_{m,n}}[k] = \begin{cases} 0\,, & se\ l = 1\ e\ m = n = 2 \\ \dfrac{1}{MN} \displaystyle\sum_{1 \le i \le M, 1 \le j \le N} ep_{l_{i,j}}[k] y_{l_{i+m-2,j+n-2}}[k]\,, & caso\ contrário \end{cases} \qquad (3.2.a)$$

$$\Delta b_{l_{m,n}}[k] = \frac{1}{MN} \sum_{1 \le i \le M, 1 \le j \le N} ep_{l_{i,j}}[k] u_{l_{i+m-2, j+n-2}}[k] \qquad (3.2.b)$$

$$\Delta c_{1_{m,n}}[k] = \frac{1}{MN} \sum_{1 \le i \le M, 1 \le j \le N} ep_{1_{i,j}}[k] y_{2_{i+m-2, j+n-2}}[k] \qquad (3.2.c)$$

$$\Delta c_{2_{m,n}}[k] = \frac{1}{MN} \sum_{1 \le i \le M, 1 \le j \le N} ep_{2l_{i,j}}[k] y_{2_{i+m-2, j+n-2}}[k] \qquad (3.2.d)$$

$$\Delta z_l[k] = \frac{1}{MN} \sum_{1 \le i \le M, 1 \le j \le N} ep_{l_{i,j}}[k] \qquad (3.2.e)$$

sendo

$$ep_{l_{i,j}}[k] = \begin{cases} \sum_{o,p \in S_R} c_{2_{o,p}} \cdot ep_{2_{o,p}} \ , & se\ l = 1 \\ \frac{1}{2}\left(d_{ll_{i,j}} - y_{l_{i,j}}[k]\right) \ , & se\ l = 2 \end{cases} \qquad (3.3)$$

onde $ep_{l_{i,j}}$ é o erro propagado da célula $Ce(i,j)$ da camada l, e $c_{2,op}$ é o operador sináptico referente à posição op da camada 2. Portanto, de acordo com (3.3), na segunda camada o erro permanecerá sendo calculado segundo (2.9) e será posteriormente utilizado em conjunto com os elementos de C_2 para o cálculo dos valores ep da primeira camada.

Contudo, mesmo com todos os ajustes descritos anteriormente, as tentativas posteriores de aprendizado de funções acopladas que exibem propagação de sinal entre as células ainda persistiam em mostrar dificuldade para obtenção de uma convergência perfeita, com ausência de erros, até mesmo nos casos com uma camada e dinâmica bipolar, onde os poucos sucessos originaram-se de situações onde o ponto inicial, escolhido aleatoriamente, avizinhava-se de uma possível solução. A presença de mínimos locais impede que a busca da solução possa percorrer uma significativa extensão do espaço do problema, exigindo um grande número de tentativas com pontos de partida diferentes para apresentar algum sucesso, tornando o CMA inadequado para ser utilizado, pelo menos isoladamente, quando se almeja uma metodologia mais abrangente e robusta. Logo, para superar tal entrave, algumas técnicas pertencentes à segunda categoria descrita na seção 2.2, as meta-heurísticas, foram analisadas. Destacaram-se como candidatos a

uma solução deste problema os Algoritmos Genéticos (GA – *genetic algorithms*), cuja grande versatilidade permitiu que seu emprego atingisse as mais diversas áreas, incluindo o aprendizado de RNA (ZHANG, DONG e XU, 2014) (BELOV e ZOLOTOV, 2015). Uma alternativa viável desta vertente para solucionar o problema em questão é a explicada na seção 2.2.2.1, cuja implementação, objeto da próxima seção, mostra uma melhor robustez.

3.1.2 Algoritmo Genético

A versão de GA proposta em (KOZEK, ROSKA e CHUA, 1993) foi realizada neste trabalho visando a aplicação em CNN multicamadas e aproveitando algumas das características utilizadas no CMA, como o modelo da CNN desenvolvido para a avaliação das possíveis soluções e a fixação de padrões das matrizes. Incluiu-se a possibilidade de escolha entre as alternativas citadas em (KOZEK, ROSKA e CHUA, 1993) para a execução das etapas de codificação, avaliação e recombinação, visando a análise quanto ao seu desempenho sob diferentes combinações.

Os valores iniciais dos indivíduos são gerados aleatoriamente seguindo uma distribuição uniforme, sendo que o número de coeficientes envolvidos dependerá da configuração estabelecida para o problema, o que engloba a seleção das matrizes que serão utilizadas e os seus padrões estruturais.

Em relação à capacidade de uso para CNN de duas camadas, as adaptações necessárias são mais simples do que para o caso do CMA. A adição da matriz C no processo se reduz a uma elevação do número de parâmetros a serem otimizados. Ademais, o fato de não prever o uso do gradiente do erro e nem a consequente retropropagação dispensa qualquer consideração quanto ao seu cálculo.

3.1.2.1 Representação real

Originalmente, aplicações de algoritmos genéticos adotavam preferencialmente a representação binária para codificação dos parâmetros a serem otimizados, aproveitando a semelhança entre um vetor de bits e uma cadeia de DNA para uma formulação intuitiva e elegante das operações genéticas envolvidas no processo, além de facilitar sua análise teórica (MICHALEWICZ, 1996).

Contudo, o surgimento de trabalhos exibindo melhores resultados ao se utilizar uma representação real promoveu um aumento deste tipo de abordagem, que

paralelamente intensificou o desenvolvimento de operadores voltados para as peculiaridades da codificação via números reais (GOLDBERG, 1991).

Além de potencialmente simplificar o processo de codificação do algoritmo e aumentar sua velocidade, esta vertente pode contribuir para aproximar a dinâmica do algoritmo ao espaço do problema, especialmente em casos que contenham variáveis pertencentes a domínios contínuos. Tal aproximação é vantajosa à medida em que pode favorecer a incorporação de operadores que explorem positivamente determinadas caraterísticas já conhecidas do problema. Na utilização de um algoritmo genético no treinamento de uma CNN pode-se, por exemplo, considerar o fato de que conjuntos diferentes de coeficientes podem corresponder a uma mesma resposta da rede, sendo conveniente a incorporação deste aspecto no funcionamento das operações genéticas.

Seguindo esta premissa, uma medida tomada neste trabalho foi a incorporação no GA da representação real para os coeficientes, contendo variantes adicionais para as operações genéticas. Os indivíduos, representantes dos coeficientes do problema, são tratados como variáveis do tipo ponto flutuante, havendo a possibilidade de limitar o número de casas decimais a partir do arredondamento realizado após as operações.

A primeira etapa para o desenvolvimento desta abordagem visou a formulação de um método de normalização dos coeficientes da CNN, aproximando valores que geram respostas similares. Partindo da equação (2.1), considerando $a_{0,0} \neq 1$ e que o estado das células esteja dentro do intervalo $[-1, 1]$, tem-se que:

$$\dot{x_{i,j}} = -x_{i,j} + a_{0,0}y_{i,j} + \left[\sum_{\substack{Ce(k,l)\in S_R(i,j),\\ k,l\neq i,j}} A(k,l)y_{k,l}\right] + \left[\sum_{Ce(k,l)\in S_R(i,j)} B(k,l)u_{k,l} + z_{i,j}\right] \quad (3.4)$$

de onde se extraiu do primeiro somatório o termo referente à saída da própria célula. Por sua vez, a equação (2.2) resulta em:

$$y_{i,j} = x_{i,j} \quad (3.5)$$

logo, no regime permanente ($\dot{x_{i,j}} = 0$), a saída da célula será dada por:

$$y_{i,j} = \frac{\left[\sum_{Ce(k,l)\in S_R(i,j),\ \ k,l\neq i,j} A(k,l)y_{k,l}\right] + \left[\sum_{Ce(k,l)\in S_R(i,j)} B(k,l)u_{k,l}\right] + z_{i,j}}{1 - a_{0,0}} \quad (3.6)$$

Sendo assim, para uma mesma combinação de sinais ($y_{k,l}$ e $u_{k,l}$), a célula responderá igualmente para dois conjuntos de coeficientes α e β que obedeçam a:

$$\frac{\rho_{i,\alpha}}{1-a_{\alpha_{0,0}}} = \frac{\rho_{i,\beta}}{1-a_{\beta_{0,0}}} \tag{3.7}$$

com ρ_i representando todos os n coeficientes presentes no numerador de (3.6). Portanto, inspirando-se nesta relação, uma forma de normalizar os coeficientes da rede consiste na divisão dos valores pelo fator $1-a_{,00}$, representada pela equação (3.8):

$$\hat{\rho}_i = \frac{\rho_i}{1-a_{0,0}} \tag{3.8}$$

sendo $\hat{\rho}$ o coeficiente normalizado.

Para o caso específico onde $a_{00}=1$, a realimentação gerada pela saída da própria célula é anulada, e a relação (3.7) perde o seu sentido. Uma alternativa viável nesta situação é a realização da normalização a partir da divisão dos valores dos coeficientes pelo somatório de seus módulos, conforme descrito por (3.9). Contudo, esta abordagem apresenta a desvantagem de não poder garantir a correspondência dos valores normalizados a uma mesma resposta da célula.

$$\hat{\rho}_i = \frac{\rho_i}{\sum_{k=1}^{n}|\rho_k|} \tag{3.9}$$

O raciocínio empregado para obtenção de (3.8) e (3.9) pode ser estendido facilmente para as células do tipo FSR. A TABELA 3.1 resume as possibilidades de normalização desenvolvidas, escritas em função do fator de normalização F_N.

TABELA 3.1. Expressões para normalização dos coeficientes.

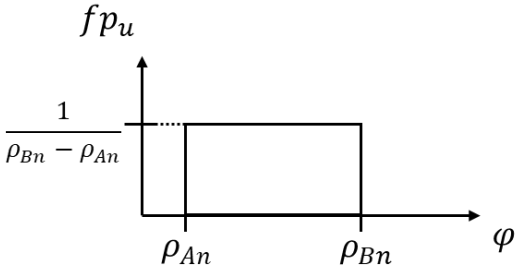
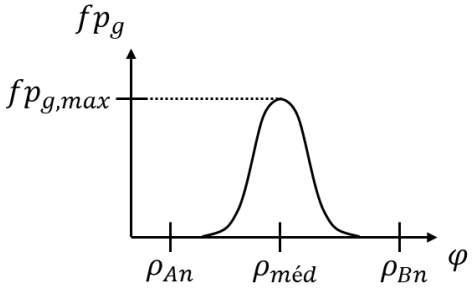
Normalização		Célula tradicional	Célula FSR
$\hat{\rho}_i = \frac{\rho_i}{F_N}$	Método 1	$\begin{cases} F_N = 1-a_{0,0}, & a_{0,0} \neq 1 \\ F_N = \sum_{i=1}^{n}\lvert\rho_i\rvert, & a_{0,0} = 1 \end{cases}$	$\begin{cases} F_N = a_{FSR,0,0}, & a_{0,0} \neq 0 \\ F_N = \sum_{i=1}^{n}\lvert\rho_i\rvert, & a_{0,0} = 0 \end{cases}$
	Método 2	$F_N = \sum_{i=1}^{n}\lvert\rho_i\rvert$	$F_N = \sum_{i=1}^{n}\lvert\rho_i\rvert$

Ambos os métodos de normalização foram incluídos como opções no fluxo do processamento do algoritmo no modo de representação real, ocupando um papel análogo às funções de codificação do caso de representação binária.

A segunda medida para a implementação do treinamento sob esta ótica, como citado anteriormente, engloba as operações genéticas às quais os indivíduos são submetidos (recombinação e mutação). Aproveitando as especificidades proporcionadas pelo uso de uma codificação real, buscou-se naturalmente abranger variadas formas de recombinar os indivíduos, baseando-se em parte nos exemplares presentes em (MICHALEWICZ, 1996). Tal busca culminou em um elenco de quatro possibilidades, listadas a seguir e ilustradas na TABELA 3.2, sendo os dois últimos formulados neste trabalho:

i. Cruzamento simples de um ponto (*One-point simple crossover*) – semelhante à alternativa binária, este método envolve a troca dos coeficientes tomando como referência um único ponto aleatório de cruzamento.
ii. Cruzamento aritmético (*Arithmetic crossover*) – gera descendentes cujos coeficientes são combinações lineares dos dois vetores envolvidos, guiadas por um parâmetro adicional a. Uma distinção adotada em relação ao cruzamento aritmético descrito em (MICHALEWICZ, 1996) é a utilização de um valor aleatório para a, provendo uma variabilidade adicional ao processo.
iii. Cruzamento com distribuição uniforme (*Uniform distribution crossover*) – dois sucessores são produzidos portando coeficientes decorrentes de uma seleção aleatória segundo uma distribuição uniforme cujos limites correspondem aos valores dos antecessores.
iv. Cruzamento com distribuição gaussiana (*Gaussian distribution crossover*) – análogo ao caso anterior, porém com a adoção de uma distribuição gaussiana, cuja média é calculada a partir dos coeficientes dos antecessores e o desvio padrão pode ser definido de duas maneiras: fixado previamente ou normalizado pela distância destes valores.

TABELA 3.2. Exemplo dos métodos de recombinação para representação real.

<table>
<tr><td></td><td colspan="2">Coeficientes</td></tr>
<tr><td>Antecessores</td><td colspan="2">$P_A = [\rho_{A1}, \rho_{A2}, \rho_{A3}, \rho_{A4}]$
$P_B = [\rho_{B1}, \rho_{B2}, \rho_{B3}, \rho_{B4}]$</td></tr>
<tr><td>Cruzamento simples de um ponto</td><td colspan="2">$P_C = [\rho_{A1} | \rho_{B2}, \rho_{B3}, \rho_{B4}]$
$P_D = [\rho_{B1} | \rho_{A2}, \rho_{A3}, \rho_{A4}]$</td></tr>
<tr><td>Cruzamento aritmético</td><td rowspan="3">$P_C = [\rho_{C1}, \rho_{C2}, \rho_{C3}, \rho_{C4}]$
$P_D = [\rho_{D1}, \rho_{D2}, \rho_{D3}, \rho_{D4}]$</td><td>$\rho_{Cn} = a * \rho_{An} + (1 - a) * \rho_{Bn}$
$\rho_{Dn} = a * \rho_{Bn} + (1 - a) * \rho_{An}$</td></tr>
<tr><td>Cruzamento com distribuição uniforme</td><td>$P(\rho_{An} < \rho_{\mathrm{Xn}} < \rho_{Bn}) = \int_{\rho_{An}}^{\rho_{Bn}} f_{p_u}(\varphi)$
 
</td></tr>
<tr><td>Cruzamento com distribuição gaussiana</td><td>$P(\rho_{An} < \rho_{\mathrm{Xn}} < \rho_{Bn}) = \int_{\rho_{An}}^{\rho_{Bn}} f_{p_g}(\varphi)$
 
</td></tr>
</table>

Quanto ao método de mutação, incorporou-se a variante não-uniforme, também explicada em (MICHALEWICZ, 1996). Assim como a versão binária da operação, cada parâmetro dos indivíduos passa por sorteios independentes que decidem a aplicação ou não do operador sobre o mesmo. A diferença no caso de representação real é que a distribuição de probabilidade utilizada para determinar seu novo valor varia com o passar das iterações do algoritmo, proporcionando uma não-uniformidade ao processo. Utiliza-se, portanto, as equações (3.10) e (3.11) para cumprir este papel.

$$\begin{cases} \rho_M = \rho + \Delta(g_a, \rho_{MÁX} - \rho), & \textit{se um dígito aleatório} \text{ é } 0 \\ \rho_M = \rho - \Delta(g_a, \rho - \rho_{MÍN}), & \textit{se um dígito aleatório} \text{ é } 1 \end{cases} \tag{3.10}$$

$$\Delta(g_a, \gamma) = \gamma \cdot (1 - r^{(1-g_a/n_g)^{\theta}}) \tag{3.11}$$

Em (3.10) e (3.11), ρ_M é o valor do parâmetro resultante da mutação, g_a é o número da geração atual, $\rho_{MÁX}$ e $\rho_{MÍN}$ são, respectivamente, o maior e o menor valor possível para o parâmetro ρ, γ é um argumento da função relacionado à distância entre o valor de ρ e um limites especificados ($\rho_{MÍN}$ ou $\rho_{MÁX}$), r é uma variável aleatória no intervalo $[0, 1]$, n_g é o número máximo de gerações do algoritmo e θ é um parâmetro do operador que determina o grau de não-uniformidade, sendo adotado neste trabalho $\theta = 0{,}5$. Desta forma, o comportamento da função Δ permite uma grande variação de ρ (partindo do intervalo $[\rho_{MÍN}, \rho_{MÁX}]$) no início do treinamento e tende a reduzir esta capacidade conforme o processo se aproxima do limite de passos (anulando-se quando $g_a \rightarrow n_g$), formando uma suave transição entre uma busca mais abrangente, em um primeiro momento, e uma mais intensa manutenção dos valores dos parâmetros quando já se espera uma sintonia mais fina. Além disso, a seleção aleatória de um dígito binário é empregada para determinar qual será o sentido desta variação. A FIGURA 3.3 ilustra o comportamento de $\Delta(g_a, \gamma)$ à medida que o treinamento avança. Nota-se que a função parte de uma característica linear decrescente na primeira geração e tende a se transformar em um impulso centrado na origem quando $g_a = n_g$.

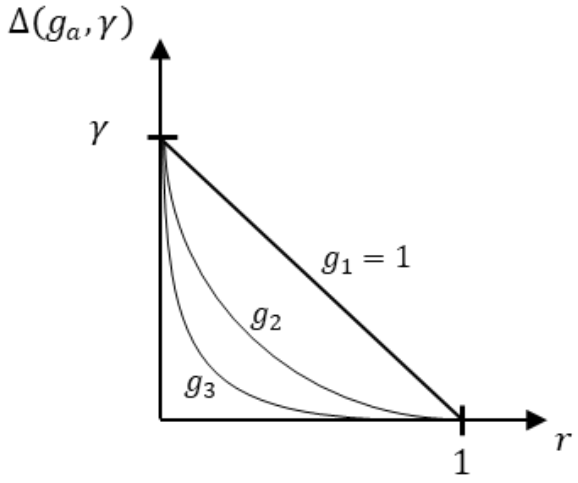


FIGURA 3.3. Fluxograma simplificado do Algoritmo Hibrido.

3.1.3 Algoritmo Híbrido

Uma terceira abordagem considerada neste trabalho aproveitou os aspectos complementares dos algoritmos estudados para uma implementação de caráter híbrido, onde a busca pelas soluções é feita primariamente por um método com capacidade mais global e complementada periodicamente por outro, mais rápido e preciso, para um refinamento local.

Com base nesta formulação, uma nova versão do algoritmo genético foi desenvolvida, englobando a possibilidade de, após cada passagem pela etapa de avaliação da simulação do modelo da CNN, uma execução do CMA ser requisitada como uma sub-rotina, tendo como ponto de partida os coeficientes representados pelo melhor indivíduo da população atual e aproveitando do GA os parâmetros de configuração aplicáveis. Este processo é ilustrado na FIGURA 3.4. A princípio, essa busca local adicional foi atribuída em todas as iterações do algoritmo. Contudo, verificou-se que desta forma o CMA poderia ser desnecessariamente executado seguidas vezes repetindo os mesmos valores iniciais, já que em certos casos o melhor indivíduo pode não apresentar mudanças durante uma sequência de gerações.

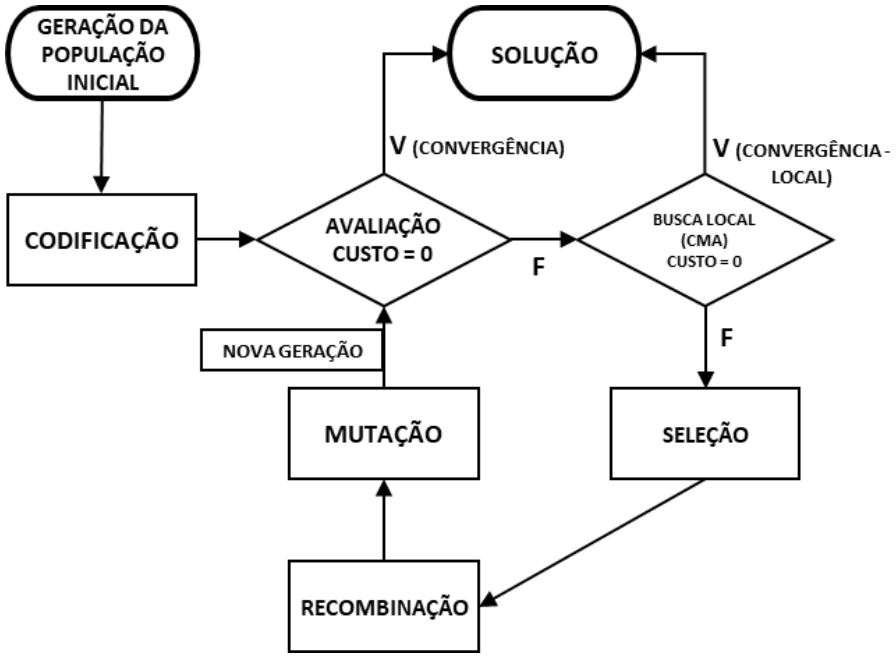


FIGURA 3.4. Fluxograma simplificado do Algoritmo Hibrido.

Por conseguinte, almejando evitar o desperdício do esforço computacional atrelado a esta situação e prover uma flexibilidade mais acentuada ao processo, adicionou-se ao algoritmo a opção de estabelecer uma frequência fixa para a utilização do

mecanismo de busca local. Uma opção adicional inserida também permite ativar este mecanismo nas situações onde se constata uma evolução genética na população, ou seja, quando o custo do melhor indivíduo é decrementado (e sua aptidão é incrementada) com a passagem de uma geração.

Ao final da execução do CMA, uma nova avaliação é efetuada sobre o seu resultado, a qual pode impactar no restante do treinamento ao obedecer duas condições: se houver convergência para uma solução válida (dentro da tolerância estipulada), esta torna-se a solução final e o treinamento é concluído; caso contrário, no cenário em que o novo custo encontrado é comparativamente menor que o original, o indivíduo é atualizado com os valores dos coeficientes fornecidos pelo CMA, ainda assim possivelmente contribuindo para acelerar a busca global.

3.2 METODOLOGIA

A metodologia de treinamento utilizada possui variações propostas visando não apenas as adaptações necessárias de algumas etapas para cada tipo de função, como também o aproveitamento de algumas de suas características previamente conhecidas. Em especial, alguns cuidados importantes devem ser considerados durante a geração de conjuntos de imagens entrada/saída. Estes e outros aspectos são tratados a seguir acompanhando estas distinções.

3.2.1 Funções Bipolares

Para as operações bipolares envolvendo CNN de uma camada, é conveniente a separação em função da existência ou não de acoplamento, em função da complexidade trazida pela capacidade de propagação de sinal intrínseca às operações acopladas. Nos casos desacoplados, tem-se que os termos da matriz *A* com exceção do termo central ($a_{0,0}$) são, por definição, nulos e, consequentemente, não possuem capacidade de propagação de sinal. Aplicando essa propriedade na equação (3.6), considerando novamente $a_{0,0} \neq 1$ e que o estado das células esteja dentro do intervalo $[-1, 1]$, obtém-se a seguinte expressão para a saída da célula em regime permanente:

$$\tilde{y}_{i,j} = \frac{\left[\sum_{C(k,l)\in S_R(i,j)} B(k,l)u_{k,l}\right] + z_{i,j}}{1 - a_{0,0}} \tag{3.12}$$

e, sendo todos os termos do lado direito da equação constantes e conhecidos, pode-se assim prever a resposta da CNN. Estendendo a análise para qualquer valor real de estado e incorporando a limitação exercida pela equação de saída (2.2), a estabilização da resposta da rede pode ser definida de forma mais geral por:

$$y_{\infty_{i,j}} = \begin{cases} 1, & \tilde{y}_{i,j} > 1 \\ \tilde{y}_{i,j}, & -1 \leq \tilde{y}_{i,j} \leq 1 \\ -1, & \tilde{y}_{i,j} < -1 \end{cases} \tag{3.13}$$

Onde $y_{\infty_{i,j}}$ denota o valor final da saída da célula $Ce(i,j)$.

E finalmente, analisando a situação específica onde $a_{00} = 1$, a equação de estado (2.1) se torna:

$$\dot{x_{i,j}} = \left[\sum_{C(k,l) \in S_R(i,j)} B(k,l) u_{k,l} \right] + z_{i,j} \tag{3.14}$$

o que corresponde a um comportamento de função bipolar onde o estado variará constantemente em um sentido definido pelo somatório do lado direito da equação, ou permanecerá inalterado caso este valor seja nulo. Sendo assim, a resposta final das células pode ser expressa por:

$$y_{\infty_{i,j}} = \begin{cases} 1, & \dot{x_{i,j}} > 0 \\ y_{0_{i,j}}, & x_{i,j} = 0 \\ -1, & \dot{x_{i,j}} < 0 \end{cases} \tag{3.15}$$

sendo $y_{0_{i,j}}$ o valor inicial da saída da célula $Ce(i,j)$.

Uma constatação obtida desta análise é que, para funções desacopladas, a resposta de cada célula dependerá apenas dos sinais de entrada na vizinhança. Por conseguinte, considerando um valor de estado inicial entre -1 e 1, condição já prevista nas aplicações tradicionais, além de uma CNN com coeficientes invariantes no espaço, todas as possibilidades de processamento serão contempladas ao se aplicar a uma mesma rede uma imagem constituída por uma concatenação de blocos com o tamanho da região de vizinhança que representem todas as combinações possíveis de sinais (pretos ou brancos). Para uma rede com raio unitário, que admite operadores conforme a FIGURA 2.4, a região de vizinhança compreende um grupo de dimensões 3x3, logo cada célula poderá receber como imagem de entrada uma combinação de 9 valores de pixels, totalizando 512

possibilidades, como exemplificado na FIGURA 3.5. Pode-se assim efetuar a implementação desse conceito a partir de criação de uma imagem de tamanho 48x96 que engloba todos esses blocos, exibida na FIGURA 3.6.

1	1	-1
-1	1	-1
-1	-1	1

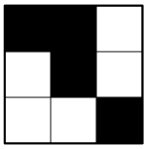


FIGURA 3.5. Exemplo de uma possível combinação de entradas para uma célula em uma CNN com $R = 1$.

FIGURA 3.6. Imagem para o treinamento de funções bipolares desacopladas.

Por outro lado, nas situações envolvendo funções acopladas e/ou que trabalham com duas camadas, a maior complexidade em sua dinâmica, representada principalmente pela capacidade de propagação do sinal, inviabiliza uma generalização da abordagem anterior. A escolha das imagens de exemplo resume-se, portanto, a incluir o máximo possível de características que podem ser relevantes ao processamento. Optou-se por um treinamento em lotes, onde um conjunto contendo um determinado número de pares de imagens é submetido, durante uma mesma iteração, à CNN, implicando em múltiplas simulações a cada passo. Visando proceder ao treinamento desta maneira, adaptações adicionais foram necessárias para ambos algoritmos: no caso do CMA, toma-se a média dos incrementos dos pesos calculados separadamente para cada par de imagens em cada iteração; já para a aplicação do GA, o cálculo do custo para cada indivíduo considera no somatório retratado pela equação (2.22) os erros dos pixels de todas as imagens envolvidas.

A aplicação do treinamento visando redes do tipo FSR pode ser realizada de forma semelhante ao descrito nesta seção, inclusive utilizando as mesmas expressões para a

atualização dos coeficientes. Uma das vantagens deste caso é que a arquitetura já limita o valor do estado dentro da faixa de trabalho dos sinais, contribuindo para a garantia de algumas considerações abordadas anteriormente. Ademais, convém ressaltar que a única distinção referente às equações (3.12) e (3.13) para este tipo de célula reside no denominador, conforme mostrado na equação (3.16), o que impacta na condição de sua validade, que se torna $a_{FSR_{00}} \neq 0$.

$$y'_{ij} = \frac{\left[\sum_{C(k,l)\in S_R(i,j)} B(k,l)u_{k,l}\right] + z_{i,j}}{a_{FSR_{00}}} \tag{3.16}$$

3.2.2 Filtros de Imagens

Em relação à abordagem da seção anterior, as principais diferenças encontradas no processo de treinamento da rede para desempenhar a ação de filtragem de imagens concernem à característica contínua da faixa de sinais, representado por tons de cinza, e à obtenção das imagens de exemplo a serem incluídas.

Tendo em vista a configuração da CNN para reproduzir o comportamento de filtros projetados no domínio da frequência, os pares de imagens entrada-saída necessários foram obtidos com base na fundamentação da seção 2.3, a partir da utilização do procedimento apresentado em (GONZALEZ e WOODS, 2007), consistindo essencialmente em cinco etapas:

i. O primeiro passo é a realização de *padding* na imagem a ser filtrada, $f(x,y)$, de dimensões $M \times N$, tal que a imagem resultante, $\bar{f}(x,y)$ possua $P \times Q$ pixels, obedecendo às relações em (2.32). Neste contexto, optou-se pela utilização de imagens quadradas ($M = N$) e admitiu-se $P = 2M$ e $Q = 2N$.
ii. Na sequência, realiza-se a centralização da transformada da imagem, multiplicando $\bar{f}(x,y)$ por $(-1)^{x+y}$, e calcular sua DFT, $\tilde{F}(u,v)$.
iii. Em seguida, efetua-se o produto entre os elementos de $\tilde{F}(u,v)$ e os da função de transferência do filtro desejado, $\tilde{H}(u,v)$, caracterizando a operação de filtragem.
iv. Com o resultado anterior, obtém-se a imagem processada $\bar{g}(x,y)$, que corresponde à imagem ampliada $\bar{f}(x,y)$, aplicando a equação (3.17), onde

somente a parte real da transformada inversa é considerada, desprezando resíduos imaginários decorrentes dos métodos numéricos envolvidos.

$$\bar{g}(x,y) = \{Re\left[\Im^{-1}\left[\tilde{G}(u,v)\right]\right\}(-1)^{x+y} \tag{3.17}$$

v. Finalmente, extrai-se a parte útil de $\bar{g}(x,y)$, que consiste do bloco de tamanho MxN formado no quadrante superior esquerdo, obtendo-se a imagem filtrada, $g(x,y)$.

vi. Para alguns tipos de filtros, como os do tipo passa-altas, pode ser necessário ainda efetuar um reescalonamento de $g(x,y)$ para adequação aos níveis de valores especificados para os pixels.

Visando examinar o máximo possível o potencial do uso da 2L-CNN para este fim, foram selecionados, para os testes, filtros com resposta ao impulso infinita (IIR), de característica mais seletiva. Tais filtros foram originados a partir da aproximação de Butterworth, contemplando-se exemplares passa-baixas e passa-altas, com as funções de transferência (2.33) e (2.35).

Um aspecto fundamental do emprego da CNN como filtro de imagens concerne às múltiplas possibilidades de aplicação de sinal (através do estado inicial e/ou das entradas) e de aproveitamento das matrizes de coeficientes sinápticos, o que se torna ainda mais amplo com a introdução da estrutura multicamadas da 2L-CNN ao cenário. Portanto, algumas convenções serão estabelecidas antecipadamente a fim de nortear o restante do processo. Os pixels da imagem de entrada serão introduzidos tanto como valores de entrada quanto como de estado inicial das células apenas da primeira camada (a matriz B_2 será nula), e a extração da imagem processada se dará pela saída da segunda camada. Por sua vez, inspirando-se no princípio da operação de filtragem no domínio espacial, descrito na seção 2.3.1, os coeficientes de limiar serão desconsiderados. Além disso, visando a aceleração da convergência do treinamento, o padrão de simetria imposto a todas as matrizes de coeficientes envolvidas é inspirado na propriedade de simetria radial presente nas classes de filtros espaciais considerados neste trabalho, correspondendo portanto ao representado por (3.18). Seguindo estas premissas, duas configurações, descritas na FIGURA 3.7, foram definidas.

$$P = \begin{bmatrix} \rho_1 & \rho_2 & \rho_1 \\ \rho_2 & \rho_3 & \rho_2 \\ \rho_1 & \rho_2 & \rho_1 \end{bmatrix} \tag{3.18}$$

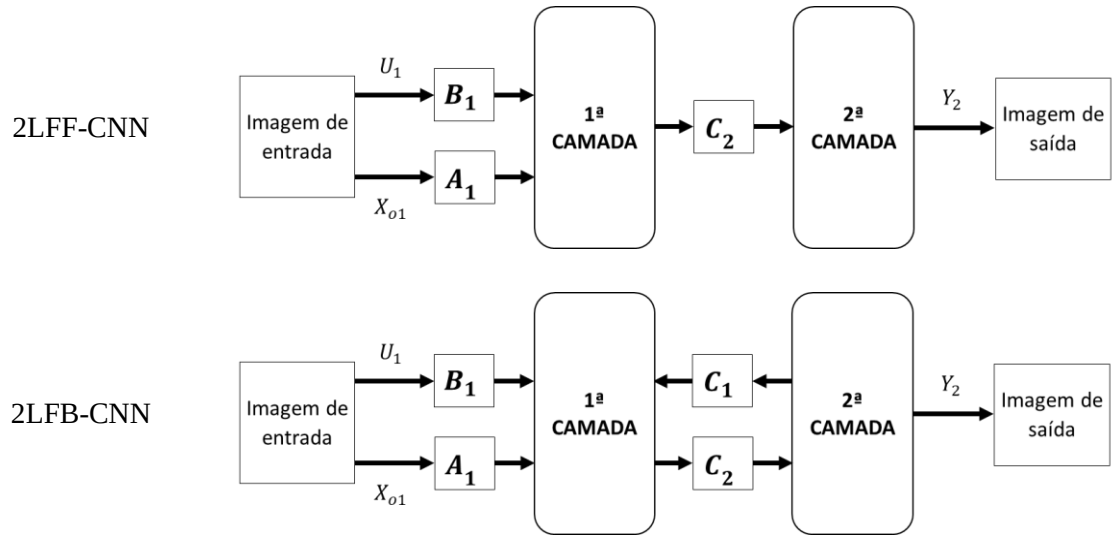


FIGURA 3.7. Configurações da CNN definidas para a filtragem de imagens.

O intervalo contínuo admitido para os valores dos pixels das imagens envolvidas nas operações de filtragem demanda uma mudança nas expectativas no processo de treinamento: não se deve esperar que os erros apresentados pelo modelo simulado da CNN, mesmo que desconsiderando a contribuição dos métodos numéricos, sejam totalmente anulados. Isto decorre do fato de que, ao contrário do que ocorre com as funções bipolares, uma reprodução exata da filtragem pela rede não pode ser, a princípio, garantida. Em vista disso, o foco da metodologia deve se voltar para a redução dos valores de erro até a busca encontrar um conjunto de coeficientes que aparente ser a melhor solução, o que pode ser indicado pela passagem de iterações que não produzam novas reduções nos erros, assumindo que não será possível ter a certeza de esta ser a melhor possibilidade no espaço de busca. Neste contexto, a sensibilidade do CMA aos pontos de mínimos locais pode pesar significativamente, e o GA, sujeito a um compromisso entre esforço computacional e precisão na busca, apresenta um maior potencial de sucesso. Similarmente ao que ocorre com as funções bipolares acopladas, um segundo ponto agravante acarretado pelo caráter contínuo dos tons de cinza da imagem consiste na inviabilidade da formação de um conjunto finito de imagens que represente completamente a dinâmica da função. Neste contexto, o emprego no treinamento de um conjunto de imagens mais numeroso ou com imagens maiores tenderá a configurar a rede para realizar a filtragem com uma capacidade mais generalista, acompanhado, todavia, de um aumento no tempo de processamento para cada iteração. Este compromisso entre

robustez e tempo valoriza a seleção de imagens que explorem o máximo possível as dinâmicas existentes na filtragem de imagens.

3.2.3 Análise da Resposta da CNN no domínio da frequência

Considerando os aspectos adicionais providos pela arquitetura em duas camadas da CNN, torna-se interessante estender o desenvolvimento realizado na seção 2.4, para uma rede simples, à 2L-CNN, possibilitando o desenvolvimento de uma análise de como o acoplamento formado pelos operadores C pode afetar a sua resposta.

Desta forma, resgatando as mesmas condições envolvidas para a 1L-CNN: estados das células limitados à faixa do sinal de saída (correspondente ao intervalo $[-1, 1]$) e matrizes de coeficientes centrossimétricas, e aplicando a transformação de Fourier discreta nas equações (2.8), referentes a uma célula do tipo FSR, tem-se que:

$$\frac{d\tilde{X}_{1_t}(\omega,\psi)}{dt} = \tilde{A}_1(\omega,\psi)\tilde{X}_{1_t}(\omega,\psi) + \tilde{B}_1(\omega,\psi)\tilde{U}_1(\omega,\psi) + \tilde{C}_1(\omega,\psi)\tilde{X}_{2_t}(\omega,\psi) + \tilde{Z}_1(\omega,\psi) \quad (3.19.a)$$

$$\frac{d\tilde{X}_{2_t}(\omega,\psi)}{dt} = \tilde{A}_2(\omega,\psi)\tilde{X}_{2_t}(\omega,\psi) + \tilde{B}_2(\omega,\psi)\tilde{U}_2(\omega,\psi) + \tilde{C}_2(\omega,\psi)\tilde{X}_{1_t}(\omega,\psi) + \tilde{Z}_2(\omega,\psi) \quad (3.19.b)$$

onde a notação com til representa a transformada de Fourier da função e o índice t indica o tempo. Diferentemente do encontrado para a 1L-CNN, as relações acima compõem um sistema de equações lineares acopladas, contendo um par de equações para cada combinação de frequências (u, v). Contudo, esta situação ainda é de menor complexidade do que é visto no domínio espacial, permitindo analiticamente a obtenção de uma solução.

Devido à dinâmica de acoplamento mútuo, a solução geral das equações (3.19) torna-se demasiadamente extensa e complexa para o propósito desta análise. Portanto, convém direcionar esta discussão para alguns casos específicos, que correspondem a respostas contidas nas aplicações de interesse. Para fins de simplificação de notação, omite-se os argumentos das transformadas, pressupõe-se $\tilde{Z}_1 = \tilde{Z}_2 = 0$ e utiliza-se o parâmetro σ dado por:

$$\sigma = \sqrt{\tilde{A}_1^2 - 2\tilde{A}_1\tilde{A}_2 + \tilde{A}_2^2 + 4\tilde{C}_1\tilde{C}_2} \quad (3.20)$$

- Caso 1: $\tilde{C}_1, \tilde{B}_1, \tilde{B}_2 \neq 0$, sendo $Re(\tilde{A}_1 + \tilde{A}_2 + \sigma) < 0$, $Re(\tilde{A}_1 + \tilde{A}_2 - \sigma) < 0$ e $\tilde{A}_1\tilde{A}_2 - \tilde{C}_1\tilde{C}_2 \neq 0$:

Esta situação é a mais ampla das analisadas aqui, empregando o acoplamento mútuo. Sua resposta final é dada por:

$$\tilde{X}_{1_\infty} = \lim_{t\to\infty} \tilde{X}_{1_t} = \frac{\tilde{C}_1\tilde{B}_2\tilde{U}_2 - \tilde{A}_2\tilde{B}_1\tilde{U}_1}{\tilde{A}_1\tilde{A}_2 - \tilde{C}_1\tilde{C}_2} \quad (3.21.a)$$

$$\tilde{X}_{2_\infty} = \frac{\tilde{C}_2\tilde{B}_1\tilde{U}_1 - \tilde{A}_1\tilde{B}_2\tilde{U}_2}{\tilde{A}_1\tilde{A}_2 - \tilde{C}_1\tilde{C}_2} \quad (3.21.b)$$

sendo $\tilde{X}_{l_\infty}$ correspondente aos valores de X da camada l em regime permanente.

- Caso 2: $\tilde{C}_1 \neq 0$; $\tilde{B}_1$, $\tilde{B}_2 = 0$, sendo $\left(\tilde{A}_1 + \tilde{A}_2 + \sigma\right) = 0$ e $Re\left(\tilde{A}_1 + \tilde{A}_2 - \sigma\right) < 0$, ou $Re\left(\tilde{A}_1 + \tilde{A}_2 + \sigma\right) < 0$ e $\left(\tilde{A}_1 + \tilde{A}_2 - \sigma\right) = 0$:

  Tais condições ainda mantêm o acoplamento mútuo, mas dispensa os operadores de entrada. A resposta da rede será, portanto,

$$\tilde{X}_{1_\infty} = \frac{\tilde{X}_{1_0}\tilde{A}_2 - \tilde{X}_{2_0}\tilde{C}_1}{\tilde{A}_1 + \tilde{A}_2} \quad (3.22.a)$$

$$\tilde{X}_{2_\infty} = \frac{\tilde{X}_{2_0}\tilde{A}_1 - \tilde{X}_{1_0}\tilde{C}_2}{\tilde{A}_1 + \tilde{A}_2} \quad (3.22.b)$$

  onde $\tilde{X}_{l_0}$ se refere aos valores iniciais de X da camada l.

- Caso 3: $\tilde{C}_1$, $\tilde{B}_1$, $\tilde{B}_2 = 0$, sendo $Re\left(\tilde{A}_1\right) < 0$ e $\tilde{A}_2 = 0$:

  Este é um caso com acoplamento em apenas uma direção, sendo o mais restrito considerado neste desenvolvimento. As expressões do estado das células podem ser derivadas das anteriores, obtendo-se:

$$\tilde{X}_{1_\infty} = 0 \quad (3.23.a)$$

$$\tilde{X}_{2_\infty} = \frac{\tilde{X}_{2_0}\tilde{A}_1 - \tilde{X}_{1_0}\tilde{C}_2}{\tilde{A}_1} \quad (3.23.b)$$

- Caso 4: $\tilde{C}_1 = 0$; $\tilde{B}_1$, $\tilde{B}_2 \neq 0$, sendo $Re\left(\tilde{A}_1\right) < 0$ e $Re\left(\tilde{A}_2\right) < 0$:

  Este cenário corresponde ao inverso do segundo caso. Logo, tem-se que:

$$\tilde{X}_{1_\infty} = -\frac{\tilde{B}_1\tilde{U}_1}{\tilde{A}_1} \quad (3.24.a)$$

$$\tilde{X}_{2\infty} = \frac{\tilde{C}_2\tilde{B}_1\tilde{U}_1 - \tilde{A}_1\tilde{B}_2\tilde{U}_2}{\tilde{A}_1\tilde{A}_2} \qquad (3.24.b)$$

A partir de uma avaliação das relações (3.21) a (3.24), percebe-se que em todas elas a dinâmica da 2L-CNN, ao menos no que concerne à sua segunda camada, convergirá, no domínio da frequência, para valores de estado resultantes de combinações lineares das entradas (casos 1 e 4) ou dos estados iniciais (casos 2 e 3) ponderadas por coeficientes. Desta forma, a transformação inversa pode ser empregada para extração dos valores de estado (e de saída) no domínio espacial. Além disso, repete-se aqui a mesma constatação citada na seção 2.4, relativa à dificuldade de formular uma correspondência entre os valores dos coeficientes nos domínios espacial e da frequência. Entretanto, restringindo-se esta ponderação para aplicações baseadas na filtragem de imagens, espera-se que a composição de termos presente nas equações anteriores possa fornecer à rede uma maior flexibilidade para atingir um desempenho mais próximo do esperado através do treinamento.

3.3 ANÁLISES COMPARATIVAS

As duas técnicas de treinamento utilizadas pertencem a duas vertentes distintas ligadas ao campo de otimização, sendo o CMA vinculado aos métodos de minimização do gradiente do erro, de caráter mais determinístico, enquanto o GA pertence à classe de meta-heurísticas, compreendendo um maior número de operações com característica estocástica. As principais distinções entre esses dois conjuntos, no que concerne ao desempenho, já são bem estabelecidas: o primeiro tende a ser em geral mais rápido, porém o segundo possui uma maior capacidade de busca da solução global. Entretanto, há alguns casos indicativos de exceções. Em (AHMAD, ISA, *et al.*, 2010), uma significativa análise de trabalhos tratando de aplicações com RNA para fins de classificação revelou exemplos onde métodos baseados no gradiente apresentaram soluções com erros menores do que algoritmos como o GA, indicando esta diferença relativa entre o desempenho das duas abordagens pode ser extremamente dependente do problema. Por conseguinte, torna-se interessante estender esse estudo para outros tipos de situações como o caso de treinamento de CNN, o que consiste na primeira análise descrita nesta seção.

Além disso, as diversas variações às quais está sujeito o GA, em especial a forma de representação dos parâmetros e os métodos de recombinação, também demandam um exame comparativo, o qual constitui o item 3.3.2.

3.3.1 CMA x GA

O estudo comparativo presente nesta seção envolve os dois métodos de treinamento adaptados no trabalho e foi documentado em (ANDRADE, SANTANA, *et al.*, 2019), onde avaliou-se a aplicação do CMA e do GA a uma CNN com uma camada e raio unitário utilizando um modelo do tipo FSR, envolvendo operações bipolares descritas em (CHUA e ROSKA, 2002) e (YANG, 2002), cujos coeficientes podem ser deduzidos analiticamente e já são conhecidos. Empregou-se a metodologia explicada na seção 3.2 e contemplaram-se casos dos tipos acoplados e desacoplados.

Para as funções desacopladas, a imagem utilizada foi a mostrada na FIGURA 3.6, já englobando todas as possibilidades de processamento. Por sua vez, para as operações acopladas, segundo o compromisso já citado na seção 3.2.2 acerca da seleção das imagens de exemplo, optou-se por um conjunto de 10 imagens de tamanhos variados, ao qual pertencem os exemplos ilustrados na FIGURA 3.8. Os coeficientes envolvidos foram selecionados para cada caso considerando o conhecimento prévio de seu funcionamento, como, por exemplo, se a imagem será inserida na rede como sinal de entrada ou de estado inicial. Também se adotou a fixação dos padrões das matrizes sempre que possível de acordo com a simetria esperada, além de se estabelecer $a_{00} = 2$ para todas as situações.

O procedimento foi realizado inteiramente por meio de código executável no ambiente do *software* Matlab®, empregando a mesma estratégia para os dois algoritmos. As especificações da plataforma computacional utilizada estão descritas na TABELA 3.3. A metodologia de repetição do treinamento foi dada da seguinte forma:

i. Para cada função, são realizadas 10 execuções independentes de treinamento.
ii. Cada execução contém um número máximo de 10 tentativas, que correspondem a um reinício do algoritmo com um novo ponto de partida aleatório, gerado a partir de uma distribuição uniforme. Além disso, cada tentativa é submetida a um limite máximo fixo de iterações, resultando em uma condição de não convergência, caso este valor seja atingido e uma solução ainda não seja encontrada.
iii. Na primeira tentativa em que o treinamento convergir para uma solução com erros absolutos abaixo da tolerância (aproximadamente zero), a execução correspondente é finalizada.

Deste modo, tanto o número de iterações em cada caso como o tempo de execução foram registrados. O algoritmo genético foi configurado utilizando uma representação binária, tomando como base alguns dos parâmetros escolhidos em (KOZEK, ROSKA e CHUA, 1993) e sendo ajustadas as suas etapas de codificação, avaliação e recombinação seguindo as configurações que levaram ao melhor resultado nesta condição. Para uma conclusão mais representativa, repetiu-se essa metodologia 10 vezes. A TABELA 3.4 lista os parâmetros especificados, baseando-se nas descrições dos algoritmos existentes na seção 2.2.

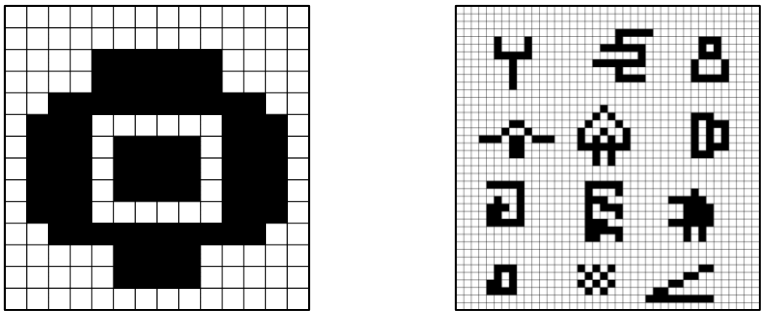


FIGURA 3.8. Exemplos de imagens utilizadas no treinamento para funções acopladas.

TABELA 3.3. Especificações do sistema utilizado para o treinamento.

Processador	Intel Core i5-3750K 4 Núcleos - 3.6 GHz
RAM	16 GB
Sistema Operacional	Windows 10 Pro x64

TABELA 3.4. Parâmetros do treinamento.

Parâmetros gerais	
Valor absoluto máximo dos coeficientes da CNN	10
Tolerância ao erro	10^{-3}
Número de tentativas	10
Número de execuções	10
Passo temporal da simulação da CNN (s)	0,1
Parâmetros do CMA	
Taxa de aprendizado (η)	10
Número máximo de iterações	1000
Parâmetros do GA	
Tamanho da população	64
Representação dos coeficientes	Binária
Tamanho do vetor binário por coeficiente	11 bits
Codificação	Aprimorada
Avaliação	Janelamento
Recombinação	Cruzamento de dois pontos
Elitismo	4
Taxa de mutação (%)	0,5
Número máximo de gerações	500

3.3.1.1 Funções Aplicadas

Esta seção apresenta uma breve descrição ilustrada das funções envolvidas, incluindo exemplos de coeficientes obtidos como solução de uma das execuções do treinamento, além de exibir imagens geradas pelo modelo da CNN sob estas configurações. Selecionaram-se 7 operações diferentes, sendo 2 desacopladas e 5 acopladas, compreendendo diferentes níveis de complexidade em relação ao número de coeficientes diferentes levados em conta.

- Detecção de bordas (*Edge*)

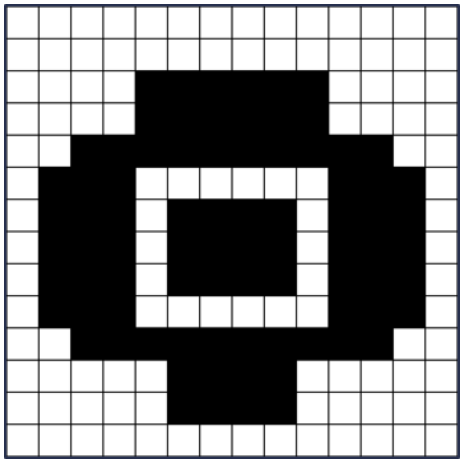
A função de detecção de bordas é um caso desacoplado dos mais simples (CHUA e ROSKA, 2002), contando com uma simetria radial e sendo reproduzida com 3 coeficientes diferentes, como pode ser visto na FIGURA 3.9.

A		
0	0	0
0	2	0
0	0	0

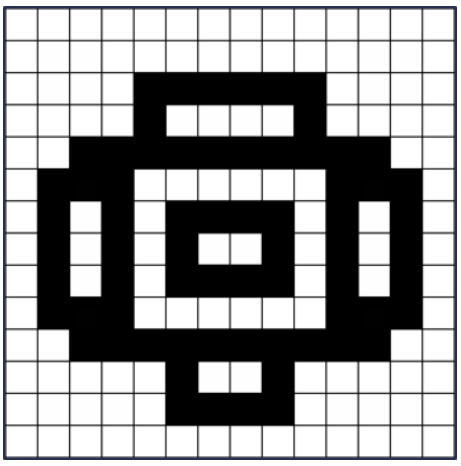
B		
b_1	b_1	b_1
b_1	b_2	b_1
b_1	b_1	b_1

$3\ coeficientes$
$33\ bits$
$\begin{cases} b_1 = -0{,}43 \\ b_2 = 9{,}39 \\ D = -6{,}63 \end{cases}$ (a)

Entrada



Saída



(b)

FIGURA 3.9. Função detecção de borda: (a) Estrutura e exemplo de resultado de treinamento; (b) Exemplo de operação produzido pelo modelo de CNN.

- Dilatação (*Dilation*)

A dilatação de objetos é um exemplo de função desacoplada que pode ser realizada de diferentes maneiras de acordo com um elemento estrutural S_d, que determinará o formato de expansão da imagem e a composição da matriz B (CHUA e ROSKA, 2002).

Devido a essa variabilidade, preferiu-se deixar seus elementos totalmente independentes, de forma que nenhum padrão fixo foi imposto a B. A FIGURA 3.10 ilustra o seu funcionamento.

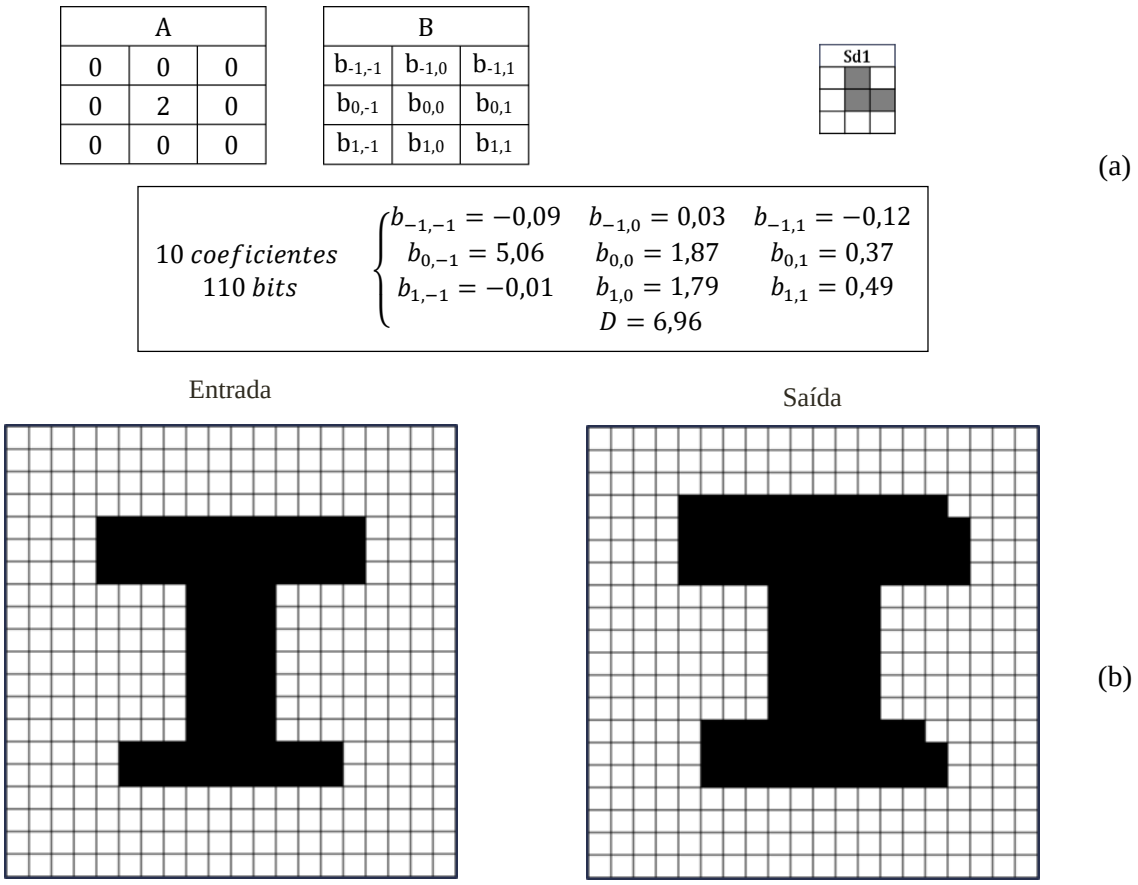


FIGURA 3.10. Função dilatação: (a) Estrutura e exemplo de resultado de treinamento; (b) Exemplo de operação produzido pelo modelo de CNN.

- Remoção de Detalhes (*Detail removing*)

Essa operação acoplada remove detalhes das imagens (grupos pequenos formados por pixels brancos adjacentes em uma região preta, ou vice-versa) (YANG, 2002). Como efeito colateral, os cantos dos objetos também são removidos, porém tal fenômeno se torna menos pronunciado visualmente quando se trabalha com imagens de grande resolução. A FIGURA 3.11 detalha suas características. Nesse caso a matriz B não é empregada e novamente optou-se por coeficientes independentes.

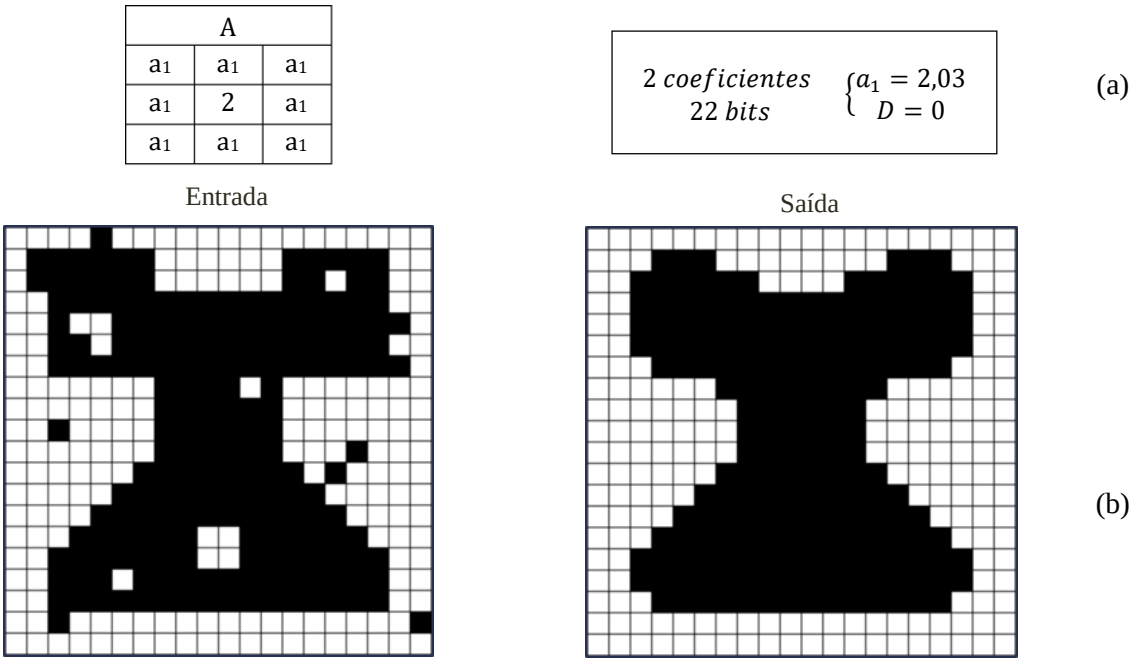


FIGURA 3.11. Função remoção de detalhes: (a) Estrutura e exemplo de resultado de treinamento; (b) Exemplo de operação produzido pelo modelo de CNN.

- Cobertura (*Covering*)

A função, também acoplada, de cobertura gera regiões pretas que conectam horizontal ou verticalmente as extremidades externas dos objetos (YANG, 2002), conforme mostrado na FIGURA 3.12. Uma peculiaridade notada é que os objetos com bordas em diagonal não sofrem alteração.

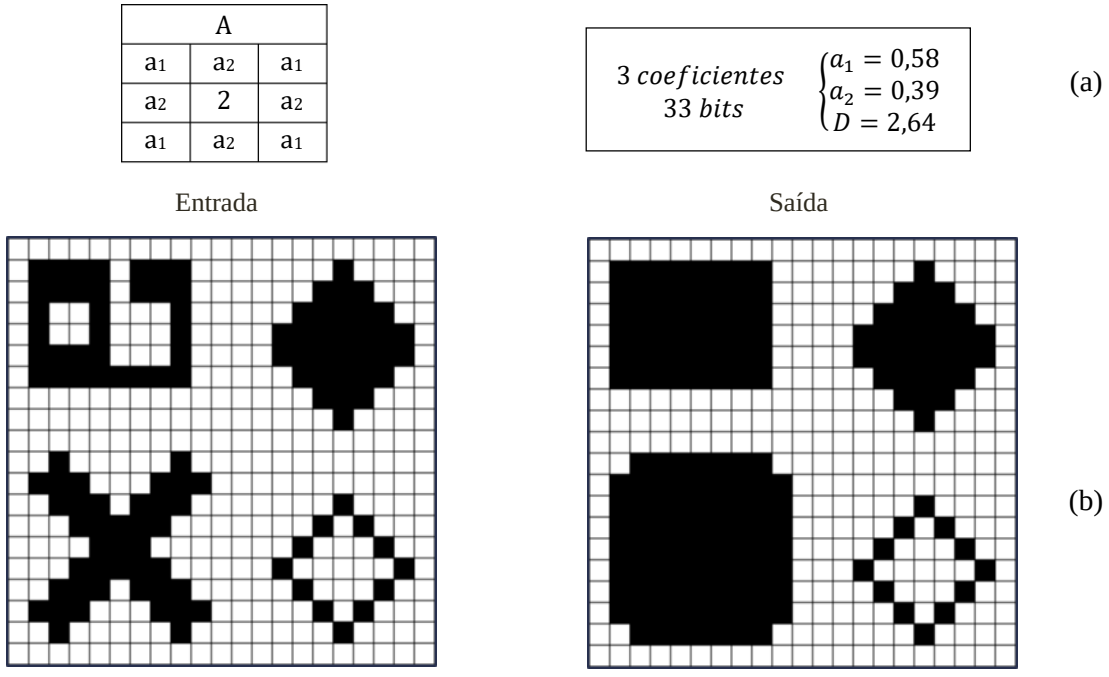


FIGURA 3.12. Função cobertura: (a) Estrutura e exemplo de resultado de treinamento; (b) Exemplo de operação produzido pelo modelo de CNN.

- Contorno Concêntrico (*Concentric contour*)

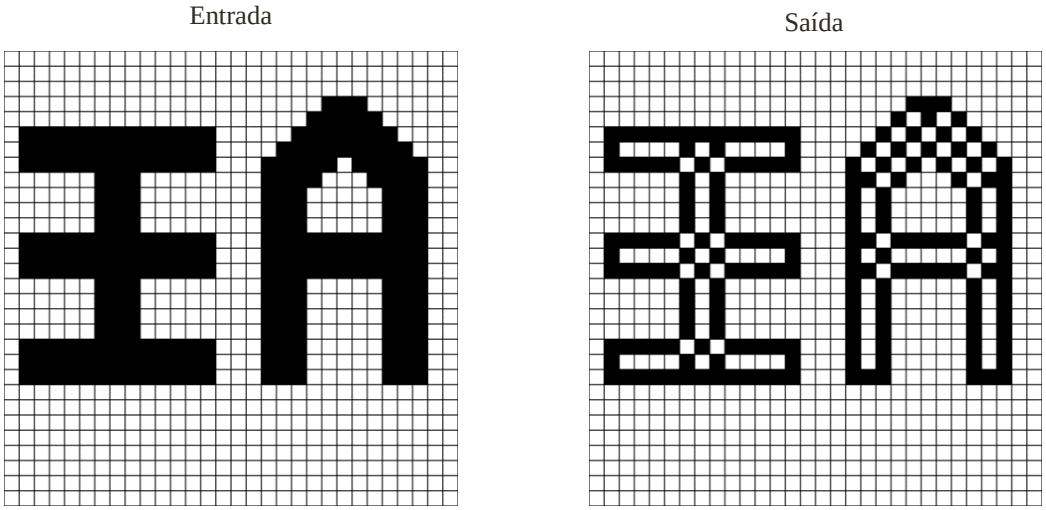
A operação acoplada de contorno concêntrico gera sobre os objetos contornos irregulares alternados pretos e brancos com características de um pixel (YANG, 2002), o que pode ser percebido observando a FIGURA 3.13.

A		
0	a_1	0
a_1	2	a_1
0	a_1	0

B		
0	b_1	0
b_1	b_2	b_1
0	b_1	0

$$4\ coeficientes \quad \begin{cases} a_1 = -1{,}75 & b_1 = -1{,}46 \\ b_2 = 9{,}08 & D = -2{,}56 \end{cases} \quad 44\ bits$$

(a)



(b)

FIGURA 3.13. Função contorno concêntrico: (a) Estrutura e exemplo de resultado de treinamento; (b) Exemplo de operação produzido pelo modelo de CNN.

- Preenchimento de Buracos (*Hole filling*)

Esta função acoplada consiste no preenchimento das regiões brancas fechadas dentro de objetos pretos (YANG, 2002). A FIGURA 3.14 mostra um exemplo de seu comportamento. Aqui já se utiliza, simultaneamente, coeficientes não centrais de ambos operadores sinápticos, com as conexões ocorrendo apenas nas direções horizontal e vertical.

- Preenchimento de fendas (*Gap filling*)

Esta operação guarda semelhanças com o preenchimento de buracos e preenche fendas brancas de um objeto, inclusive as que possuem aberturas, tornando-as pretas. Contudo, nesse caso o objeto em si não será incluído ao resultado (YANG, 2002). O processo pode ser visualizado na FIGURA 3.15.

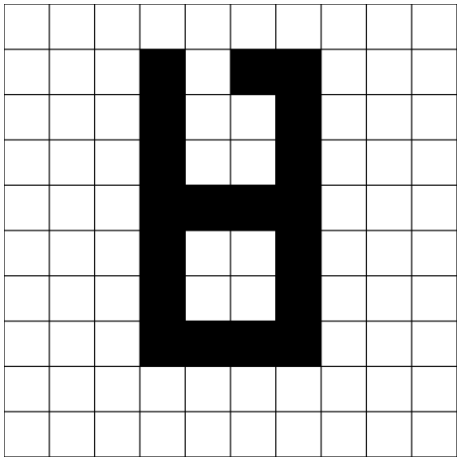
A		
0	a_1	0
a_1	2	a_1
0	a_1	0

B		
0	b_1	0
b_1	b_2	b_1
0	b_1	0

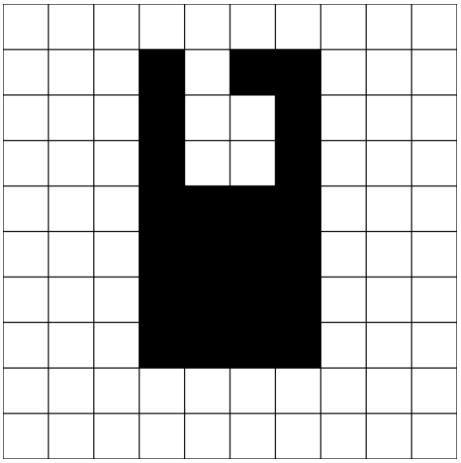
$$4\ coeficientes \quad \begin{cases} a_1 = 1{,}25 & b_1 = 0{,}24 \\ b_2 = 8{,}42 & D = 3{,}36 \end{cases} \quad 44\ bits$$

(a)

Entrada



Saída



(b)

FIGURA 3.14. Função preenchimento de buracos: (a) Estrutura e exemplo de resultado de treinamento; (b) Exemplo de operação produzido pelo modelo de CNN.

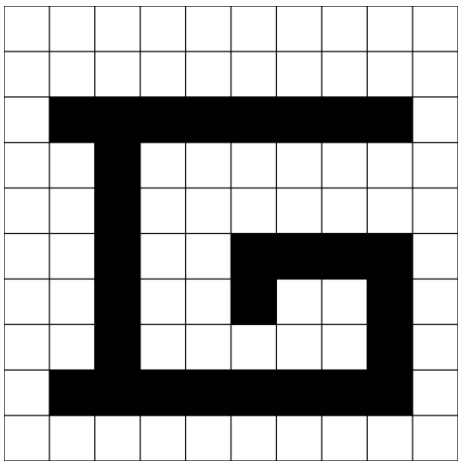
A		
0	a_1	0
a_1	2	a_1
0	a_1	0

B		
0	b_1	0
b_1	b_2	b_1
0	b_1	0

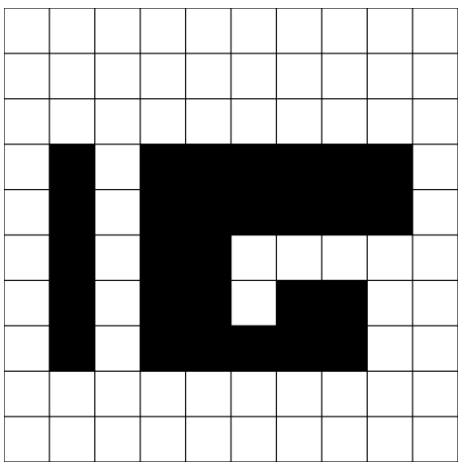
$$4\ coeficientes \quad \begin{cases} a_1 = 2{,}17 & b_1 = 2 \\ b_2 = -7{,}98 & D = -4{,}73 \end{cases} \quad 44\ bits$$

(a)

Entrada



Saída



(b)

FIGURA 3.15. Função preenchimento de fendas: (a) Estrutura e exemplo de resultado de treinamento; (b) Exemplo de operação produzido pelo modelo de CNN.

3.3.1.2 Análise de Desempenho

A TABELA 3.5 resume os resultados obtidos, exibindo algumas figuras de mérito para a avaliação das duas técnicas. O tempo total de execução e o número de simulações da CNN efetuados são expressos pelo seu valor médio e seu desvio padrão e englobam todas as 10 execuções independentes, mesmo aquelas onde a convergência não foi atingida. Uma ponderação importante decorre do fato de a maior parte da carga computacional do treinamento se originar na etapa de avaliação do erro, onde um modelo da rede é simulado, o que se traduziria a princípio numa forte correlação entre estas duas métricas. Estendendo este raciocínio para os dois algoritmos, a quantidade de execuções da CNN corresponde, no caso do CMA, ao produto entre número de iterações e a quantidade de imagens envolvidas, enquanto que, no GA, este valor carrega ainda um fator multiplicativo equivalente ao tamanho da população. Logo, para cada passo nas mesmas condições, esperar-se-ia que o GA levasse em média um tempo de processamento igual ao produto entre o tamanho da população e o tempo correspondente ao CMA, com ambos mantendo, contudo, uma proporcionalidade similar entre o número de execuções e o tempo total. Todavia, o que se pôde observar nas razões presentes na tabela, que relacionam ambas as métricas para cada algoritmo, é a invalidação desta afirmação, possivelmente provocada por fatores de variabilidade como:

i. A escolha aleatória do ponto de partida;
ii. Os diferentes tempos de acomodação das simulações da CNN;
iii. O número de iterações necessárias em cada execução;
iv. Os ganhos de eficiência dos algoritmos decorrente de otimizações e do paralelismo empregado pelo *software*;
v. As flutuações no poder de processamento do sistema computacional disponível para o *software* de treinamento, que concorre com outros processos.

Tais aspectos também contribuem para a grande dispersão revelada pelos indicadores estatísticos relacionados a estas figuras.

Elementos adicionais incluídos na TABELA 3.5 são o número de tentativas médias por corrida e duas taxas de sucesso diferentes: a taxa de sucesso de execuções corresponde à proporção de execuções onde uma das possíveis dez tentativas produziu

um resultado com convergência sem erros, ignorando, portanto, o número de tentativas realizadas; já a taxa de sucesso de tentativas advém da frequência de convergência considerando-se todas as tentativas realizadas nas dez execuções.

TABELA 3.5. Parâmetros de desempenho do treinamento.

Função		**Algoritmo**	**Tempo total (s)**			**Execuções da CNN**			**Média de tentativas**	**Taxa de sucesso de execuções (%)**	**Taxa de sucesso de tentativas (%)**
Categoria	*Nome (Nº de pesos)*		*Média*	*Desvio*	*Razão GA/ CMA*	*Média*	*Desvio*	*Razão GA/ CMA*			
Desacoplada (1 imagem)	Detecção de borda (3)	CMA	12,71	8,88	15,12	60	41,01	328,33	1	100	100
		GA	192,124	561,44		$1{,}97 \cdot 10^4$	$4{,}33 \cdot 10^4$		1,3	100	76,92
	Dilatação (10)	CMA	10	2,11	1,19	10	2,11	249,60	1	100	100
		GA	11,87	3,49		2496	662,36		1	100	100
Acoplada (10 imagens)	Remoção de detalhes (9)	CMA	200,05	240,47	6,01	7576	$8{,}91 \cdot 10^4$	26,40	2,1	100	47,62
		GA	1202,17	1394,1		$2 \cdot 10^5$	$2{,}78 \cdot 10^5$		1,1	100	90,91
	Cobertura (3)	CMA	69,96	74,46	2,53	2588	2785,68	21,52	1,3	100	76,92
		GA	176,98	95,49		$5{,}57 \cdot 10^4$	$3{,}10 \cdot 10^4$		1	100	100
	Preenchimento de fendas (4)	CMA	2010,87	189,22	0,02	$4{,}9 \cdot 10^4$	3133,82	0,60	9,9	10	1,01
		GA	37,28	27,84		$2{,}95 \cdot 10^4$	$2{,}64 \cdot 10^4$		1	100	100
	Preenchimento de buracos (4)	CMA	1350,22	802,84	0,02	$3{,}46 \cdot 10^4$	$1{,}99 \cdot 10^4$	0,76	7,4	50	6,76
		GA	32,43	20,93		$2{,}62 \cdot 10^4$	$1{,}66 \cdot 10^4$		1	100	100
	Contorno concêntrico (4)	CMA	1432,58	855,89	0,04	$3{,}48 \cdot 10^4$	$2{,}07 \cdot 10^4$	2,03	7,3	50	6,84
		GA	52,02	46,07		$7{,}08 \cdot 10^4$	$6{,}49 \cdot 10^4$		1	100	100

A princípio pode-se notar um melhor desempenho do CMA nos casos de funções desacopladas em todos os aspectos por uma boa margem, obtendo sucesso total com apenas uma tentativa. Esta ocorrência leva à confirmação de que os métodos baseados no gradiente são realmente mais que suficientes para o aprendizado deste tipo de função, cuja dinâmica é mais simples. Soma-se a isso o fato de o GA falhar em duas tentativas durante o treinamento da operação de detecção de borda, destacando que a presença de aleatoriedade em seu funcionamento pode levar a uma falta de convergência mesmo nas situações relativamente mais triviais.

Em relação aos modelos de funções acopladas, é notável que a eficiência do CMA é reduzida. Embora este algoritmo ainda apresente bons resultados no caso da função de cobertura, os demais resultados apresentam taxas de convergência abaixo de 50%, não

chegando nem a 10% nas três últimas funções da TABELA 3.5. As taxas de sucesso das execuções também são afetadas, deixando de alcançar 100%. O GA, por outro lado, mantém sua grande capacidade de pesquisa como esperado, ficando aquém de seu objetivo em apenas uma tentativa, especificamente para a operação de remoção de detalhes.

Outro aspecto importante observado nesta análise é a maior velocidade do CMA na maioria dos casos, como previsto. Para duas das funções acopladas, a convergência levou apenas um terço do tempo do GA. Para os outros três casos de funções desacopladas, entretanto, ocorre o contrário, devido ao fato de o tempo total do treinamento ter sido altamente influenciado pelas tentativas malsucedidas, mais frequentes nessas situações, o que torna o GA a única técnica viável e corrobora com a afirmação de (AHMAD, ISA, *et al.*, 2010) segundo a qual aspectos relacionados às diferentes aplicações, ou operações, podem influenciar criticamente na subversão das vantagens tradicionalmente atribuídas a cada técnica de treinamento.

3.3.2 Métodos de Recombinação

O segundo ensaio tratado neste capítulo compreende a análise da etapa de recombinação do GA, que se mostrou ser a mais influente no funcionamento do treinamento, principalmente pelo grande número de variantes implementadas. Desta forma, realizou-se diversas execuções de algumas das funções bipolares tratadas na seção 3.3.1 empregando-se as diferentes técnicas de recombinação, incluindo as duas possibilidades de representação dos coeficientes. Para reduzir a influência da operação de mutação nesta análise, sua taxa foi mantida em um valor baixo. Aproveitou-se a situação para também incluir dois cenários representando a abordagem híbrida de treinamento, sendo que em um deles a taxa de mutação é aumentada, visando estudar a sua contribuição.

A metodologia utilizada permaneceu bem similar à análise anterior, com apenas duas distinções relevantes, visando uma maior uniformidade. A primeira se refere à alteração do conjunto de imagens considerado para todas as funções, que contou com a inclusão de exemplares maiores e/ou mais complexos, resultando em um novo conjunto de 8 imagens. Já a segunda modificação corresponde à forma de repetição do treinamento: para cada cenário proposto, contabilizou-se 100 execuções independentes. A TABELA 3.6 descreve os parâmetros do treinamento comuns a todos os cenários, enquanto a TABELA 3.7 lista suas caraterísticas diferenciadas.

TABELA 3.6. Parâmetros gerais do treinamento pelo GA.

Parâmetros gerais	
Valor absoluto máximo dos coeficientes da CNN	10
Tolerância ao erro	10^{-3}
Número de tentativas	100
Passo temporal da simulação da CNN (s)	0,1
Número de casas decimais dos parâmetros	2
Tamanho da população	64
Avaliação	Janelamento
Elitismo	3
Número máximo de gerações	200

TABELA 3.7. Cenários de treinamento.

<table>
<tr><th rowspan="2">Cenário</th><th rowspan="2">Algoritmo</th><th colspan="4">Parâmetros</th></tr>
<tr><th>Representação</th><th>Codificação/ Normalização</th><th>Taxa de Mutação (%)</th><th>Recombinação (Cruzamento)</th></tr>
<tr><td>B-1</td><td rowspan="7">GA</td><td rowspan="2">Binária</td><td rowspan="2">Codificação aprimorada</td><td rowspan="7">0,5</td><td>Dois pontos</td></tr>
<tr><td>B-2</td><td>Aleatório</td></tr>
<tr><td>R-1</td><td rowspan="6">Real</td><td rowspan="6">Método de normalização 2</td><td>Aritmético</td></tr>
<tr><td>R-2</td><td>Distribuição uniforme</td></tr>
<tr><td>R-3</td><td>Distribuição gaussiana (desvio fixo)</td></tr>
<tr><td>R-4</td><td>Distribuição gaussiana (desvio normalizado)</td></tr>
<tr><td>H-1</td><td rowspan="2">Híbrido</td><td rowspan="2">Distribuição gaussiana (desvio fixo)</td></tr>
<tr><td>H-2</td><td>10</td></tr>
</table>

Os resultados são resumidos na TABELA 3.8. Para cada função, os cenários são ordenados segundo as taxas de sucesso geral, que representam a frequência de convergência, de forma decrescente. Ademais, estão presentes as taxas de sucesso associada às ocorrências em que a busca local encontra a solução, aplicáveis apenas aos cenários englobando o algoritmo híbrido. A princípio, nota-se que as taxas atingem, em sua maioria, faixas diferentes de acordo com a função, como já indicado na análise da seção 3.3.1. Neste sentido, os padrões se repetem, com o treinamento das funções desacopladas tendo bem mais êxitos do que as funções acopladas, dentre as quais destaca-se a remoção de detalhes como a mais difícil para a obtenção de uma solução.

TABELA 3.8. Parâmetros de desempenho do treinamento

Função			**Cenário**	**Taxa de sucesso - geral (%)**	**Taxa de sucesso - busca local (%)**	**Nº de gerações**		**Tempo total (s)**		**Tempo médio por geração (s)**
Categoria	*Nome*	*Nº de pesos*				*Média*	*Desvio*	*Média*	*Desvio*	
Desacoplada	Detecção de borda	3	H-2	100	33	5,18	8,21	2,84	2,92	0,55
			R-3	93	-	26,90	57,01	8,39	17,45	0,31
			H-1	93	46	20,90	52,18	7,71	14,91	0,37
			R-1	87	-	38,08	69,38	14,53	27,67	0,38
			R-2	81	-	51,62	78,51	15,71	23,75	0,30
			R-4	81	-	55,58	79,97	16,25	23,20	0,29
			B-2	79	-	50,82	80,49	16,01	25,24	0,31
			B-1	65	-	80,26	92,74	24,79	28,60	0,31
	Dilatação	10	B-1	100	-	51,80	21,01	20,18	7,87	0,39
			B-2	100	-	44,18	9,61	15,92	3,46	0,36
			R-3	100	-	35,70	15,21	14,07	6,02	0,39
			H-1	100	36	6,88	4,47	23,88	10,89	3,47
			H-2	100	35	12,00	9,88	26,60	14,51	2,22
			R-2	99	-	54,31	36,30	19,46	12,91	0,36
			R-4	98	-	104,65	41,85	39,31	15,73	0,38
			R-1	96	-	102,07	41,44	36,65	14,89	0,36
Acoplada	Remoção de detalhes	9	H-2	25	6	169,66	61,17	125,78	39,74	0,74
			H-1	20	7	168,6	63,89	102,91	35,37	0,61
			R-3	19	-	170,85	61,42	91,43	30,94	0,54
			B-1	13	-	184,8	43,94	93,7	24,08	0,51
			R-1	12	-	179,97	54,97	92,13	25,83	0,51
			R-2	10	-	182,21	53,8	93,31	26,06	0,51
			B-2	5	-	193,4	29,45	93,2	15,37	0,48
			R-4	4	-	195,68	24,66	100,29	18,03	0,51
	Preenchimento de fendas	4	H-2	100	20	54,84	40,63	123,13	70,70	2,25
			B-2	34	-	142,50	82,76	95,52	57,55	0,67
			H-1	28	2	177,19	50,88	326,12	107,67	1,84
			R-3	21	-	178,46	50,51	146,46	50,78	0,82
			B-1	19	-	169,84	66,27	136,65	83,75	0,80
			R-2	11	-	184,12	48,41	141,28	75,17	0,77
			R-1	9	-	187,66	44,86	146,83	93,22	0,78
			R-4	9	-	185,73	46,34	111,97	58,26	0,60

Por sua vez, explorando as diferenças entre os cenários, destaca-se que a presença mais significativa da mutação em H-2 o coloca sempre na primeira posição em termos de sucesso, apesar de empatar com outros quatro cenários no caso da dilatação, com uma taxa de 100% de sucesso. A mutação, inclusive, mostra-se bem útil para o preenchimento de fendas, contribuindo acentuadamente para a convergência ocorrer em todas as tentativas em H-2.

A análise das taxas geradas por H-1 permite verificar os ganhos gerados pela abordagem híbrida, principalmente numa comparação com R-3, cenário este que também utiliza o cruzamento com distribuição gaussiana (com desvio padrão fixo) mas não emprega a busca local. Percebe-se que, enquanto o desempenho de ambos se manteve bem similar para os casos desacoplados, houve uma melhora da frequência de sucesso nas outras funções. Na situação específica da operação de remoção de detalhes, a busca local foi responsável diretamente por mais de um terço dos sucessos, fornecendo indícios positivos acerca da proposta de aproveitar uma combinação do GA e do CMA para otimizar o processo.

No que concerne às técnicas de recombinação, as figuras de desempenho exibem uma certa alternância. Contudo, é possível visualizar uma tendência de vantagem para o cenário R-3, que envolve o cruzamento com distribuição gaussiana e desvio padrão normalizado. De fato, dentre os cenários R e B, o mesmo apenas não liderou a comparação na função de preenchimento de fendas. Este é um indicativo do potencial do uso da representação real no GA, e da importância da flexibilidade que o mesmo provê na formulação das operações genéticas que sejam mais adequadas à aplicação em questão.

As demais figuras presentes na TABELA 3.8, especificamente o tempo e o número de gerações levados pelos algoritmos, permitem constatações adicionais. Nos cenários com o algoritmo híbrido, a redução na quantidade média de gerações se deve à antecipação da convergência promovida pelo CMA, estando, de forma lógica, bem correlacionada com a taxa de sucesso da busca local. Ainda assim, nestes casos o tempo médio por geração atingiu, em geral, níveis mais acentuados. A explicação deste fenômeno está associada à elevada duração relativa das execuções do CMA, que, ao não contar com o mesmo grau de paralelismo presente no GA, impacta de forma acumulativa no processo. Para contornar esta situação, pode ser interessante ajustar a frequência da aplicação do CMA, cuja ativação, neste estudo, foi dada sempre que houvesse evolução entre as gerações, buscando um compromisso que reduza o tempo levado pelo treinamento, mas ainda assim permita uma execução significativa da busca global. Isto

pode ser também auxiliado pelo ajuste do número máximo de passos de cada execução do CMA (considerado 30 nesta análise). Além disso futuras otimizações do próprio algoritmo podem contornar estas limitações de velocidade.

4 CNN ANALÓGICA DE DUAS CAMADAS

Neste capítulo, é abordada a realização de uma CNN analógica de duas camadas em tecnologia CMOS, a partir de uma rede simples, com destaque para as adaptações introduzidas no projeto. Também são apresentados os resultados de simulação deste circuito no processamento de imagens bipolares e na filtragem de imagens em escala de cinza.

4.1 REALIZAÇÃO

O modelo de célula de CNN desenvolvido em (SANTANA, FREIRE e CUNHA, 2012), que foi utilizado neste trabalho, foi aplicado satisfatoriamente em uma grande gama de funções realizadas pela CNN simples (com apenas uma camada). Como uma evolução natural, pode-se pensar em sua extensão visando a estrutura de duas camadas descrita na seção 2.1.1. Sendo assim, neste trabalho, procedeu-se à realização dessa nova rede, adicionando os blocos necessários, mais especificamente, os multiplicadores responsáveis pelas sinapses das conexões intercamadas. Contudo, o aumento do número de sinais de corrente representando esses acréscimos afetou acentuadamente o desempenho nas simulações, à medida que as imperfeições de cada multiplicação individual se acumularam de forma mais intensa, ampliando a possibilidade da ocorrência de erros na operação da rede. Após a realização de uma análise mais aprofundada do fenômeno, foram percebidos alguns fatores que prejudicaram o funcionamento, demandando, portanto, a introdução de medidas para compensação.

Para as simulações descritas neste capítulo os coeficientes das operações foram convertidos, a princípio, para sinais de corrente seguindo a relação de 25 nA por unidade. Posteriormente, para acelerar o processamento da rede e melhorar o desempenho em algumas situações, procurou-se elevar proporcionalmente seus valores, estabelecendo um limite de forma que o(s) coeficiente(s) mais alto(s) ainda se mantivessem dentro da faixa em que o grau de linearidade ainda é satisfatório. Desta forma, o maior nível empregado correspondeu a um nível de corrente absoluto de 350 nA, região de operação esta em que os multiplicadores presentes no sistema ainda exibem baixa distorção harmônica. Os sinais referentes às entradas e aos estados das células foram mapeados segundo uma correspondência linear onde os pixels branco e preto equivalem aos valores de tensão de -15 mV e 15 mV, respectivamente. O circuito, composto por uma versão da 2L-CNN com dimensões de 10×10 células em cada camada, foi simulado na forma pré-leiaute no

ambiente do pacote de ferramentas da Mentor Graphics®, que utiliza o simulador ELDO, disponível no Laboratório de Concepção de Circuitos Integrados (LCCI) da Escola Politécnica da Universidade Federal da Bahia (UFBA). Os resultados foram comparados àqueles gerados pelo mesmo modelo da rede utilizado no treinamento, descrito no código executável no ambiente do *software* Matlab®.

4.1.1 Aprimoramentos da CNN

A seguir são descritas algumas limitações do circuito CMOS projetado para a realização de células de CNN em (SANTANA, et al., 2012) e revisitado na seção 2.5 deste trabalho. Tais limitações foram toleradas na operação da CNN com uma camada, mas tornaram-se críticas para a realização da CNN de duas camadas. Assim, visando mitigar cada uma destas limitações, implementou-se algumas modificações, seja na arquitetura do circuito, seja no ajuste de operadores sinápticos, também explanados nos itens que seguem.

4.1.1.1 Assimetria dos multiplicadores

Notou-se que os blocos multiplicadores que atuam como sinapses no circuito de (SANTANA, FREIRE e CUNHA, 2012), realizando o produto entre um sinal da célula e um coeficiente da função, exibem respostas assimétricas em relação à polaridade do sinal nas entradas. De fato, os valores de coeficientes negativos levam a um resultado maior em termos absolutos do que seus opostos, conforme os resultados de simulação mostrados na FIGURA 4.1. Tal aspecto foi atenuado adotando-se um fator multiplicativo k_1 menor que um, determinado empiricamente a partir da análise das características da FIGURA 4.1, para ajuste dos coeficientes negativos, conforme a relação $\rho_a = k_a \rho$, onde ρ_a corresponde ao parâmetro ρ ajustado e $k_a = 0{,}91$.

4.1.1.2 *Offset* em uma das entradas dos multiplicadores

Como explicado na seção 2.5, o circuito multiplicador concebido para a realização das sinapses em (SANTANA, et al., 2012) possui uma entrada em tensão e uma entrada em corrente, além da saída também ser um sinal de corrente proporcional ao produto dos sinais de entrada. Contudo, limitações físicas do circuito, ocasionados por pequenas assimetrias ou descasamentos de parâmetros, produzem um deslocamento na corrente de saída originado por um pequeno nível de corrente constante (*offset*) verificado na entrada

em corrente mesmo quando o sinal de entrada é nulo. Este deslocamento é ilustrado na FIGURA 4.2. Apesar de relativamente pequeno quando comparado aos níveis de corrente que representam os coeficientes sinápticos tipicamente envolvidos (da ordem de dezenas ou centenas de nA), ele pode interferir fortemente nos casos em que um ou mais coeficientes são nulos, à medida em que as respostas das sinapses correspondentes produzem um valor diferente do esperado (zero). Tal fato se sobressai mais claramente sob algumas condições, como por exemplo quando os sinais de tensão envolvidos possuem valor absoluto elevado ou em funções bipolares onde combinações de sinais que levam a respostas opostas correspondem a níveis bem próximos no somatório das sinapses. Sendo assim, este pequeno *offset* pode ser suficiente para mudar o sentido da variação do estado, contribuindo para um desvio do comportamento da CNN. Este efeito pode ainda ser amplificado com o aumento no número de pesos nulos presentes na função. Para contornar tal situação, substituíram-se as correntes que representam os coeficientes nulos pelo valor que produz um sinal nulo, encontrado empiricamente de tal forma que se $\rho = 0$, então $\rho_a = I_Z = 1{,}14\ nA$.

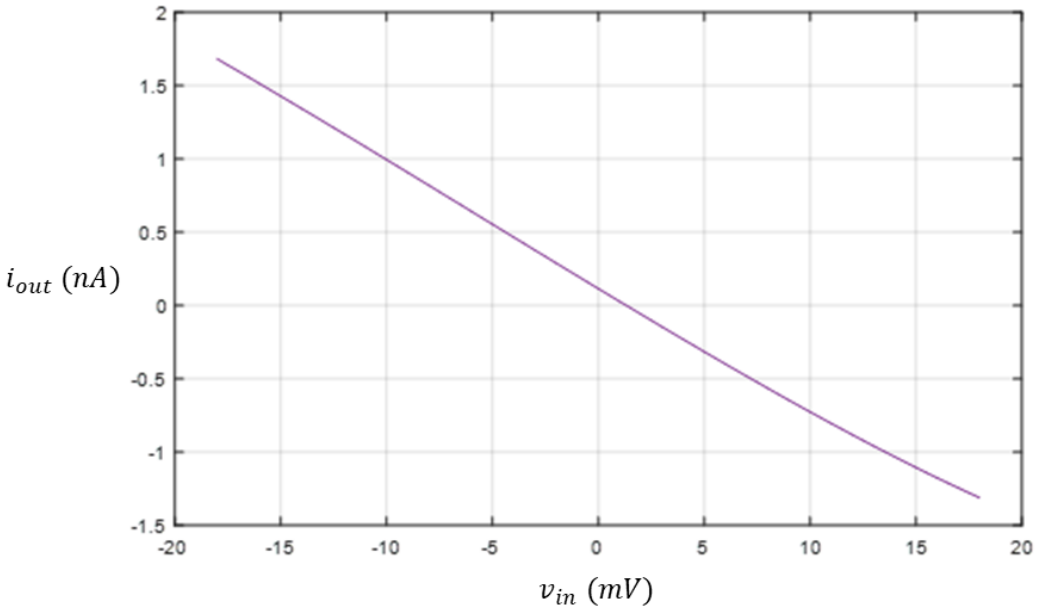


FIGURA 4.1. Característica DC do multiplicador da sinapse para $i_{in} = 0$.

4.1.1.3 Sobressinal do regime transitório do estado das células

No início do processamento da rede, sua resposta transitória apresentava um elevado sobressinal, atingindo na maioria das vezes níveis além do estabelecido para a operação do circuito. Mesmo que temporário, este comportamento possui o potencial de alterar significativamente o estado final do sistema, sobretudo nas situações em que se tem a condição $a_{00} > 0$ para células do tipo FSR, onde o valor positivo do coeficiente produz uma realimentação positiva. O meio escolhido para evitar o problema foi a adição de um elemento capacitivo no terminal de saída da célula, amortecendo sua variação de tal forma que seja mantida dentro do permitido para o funcionamento adequado. A contrapartida desta inclusão é o aumento de tempo de processamento. A FIGURA 4.3 ilustra um exemplo onde se avalia a resposta em simulação para três possibilidades. Através de ajustes empíricos deste elemento capacitivo em testes com algumas operações mais críticas, observou-se uma boa taxa de sucesso com um valor de capacitância de 50 pF, sendo então incorporado ao circuito nas demais simulações.

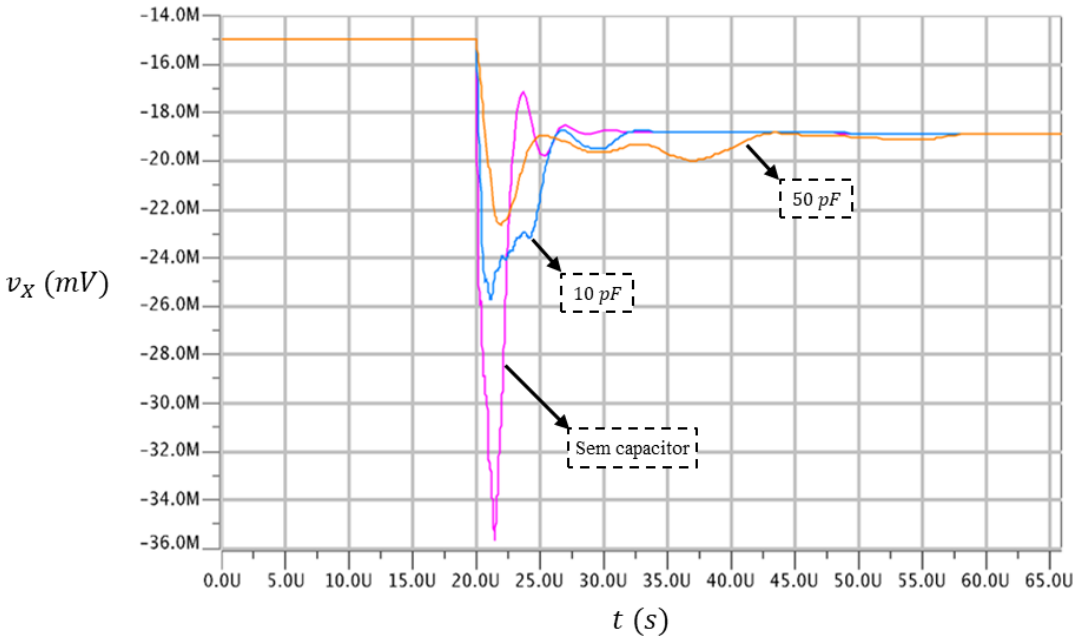


FIGURA 4.2. Resposta transiente de uma célula da CNN para 3 casos: sem capacitor, capacitor de 10 pF e capacitor de 50 pF.

4.1.1.4 Limitação do grampeamento

Outra origem de efeitos negativos ao funcionamento da CNN reside em limitações do circuito grampeador. A arquitetura utilizada originalmente, proposta em (HEGT, LEENAERTS e WILMANS, 1998), possui uma característica de resposta DC cujas

inclinações nas regiões limites, que no caso ideal correspondem a retas verticais, não são suficientemente elevadas, o que resulta em situações onde, mesmo após a estabilização, a tensão que representa o estado das células pode ultrapassar significativamente a faixa de operação especificada. De fato, em algumas simulações contendo células com uma corrente total elevada, resultante das sinapses, o estado poderia atingir valores absolutos próximos a 19 mV, portanto significativamente superior ao valor absoluto estabelecido como correspondente aos pixels branco ou preto (15 mV). Por outro lado, a realimentação presente no bloco sináptico cujo estado atinge valor absoluto desta grandeza pode gerar uma corrente de saída com magnitude muito superior à prevista, o que pode afetar criticamente o processamento. Neste contexto, uma característica DC corrente-tensão de grampeamento com transições mais abruptas nos extremos contribuirão para manter os valores de estado extremos mais próximos do especificado. Seguindo este raciocínio, buscou-se substituir o circuito grampeador por uma alternativa que apresentasse tal comportamento, sendo o circuito escolhido ilustrado na FIGURA 4.4, que utiliza pares de inversores MOS para controlar a corrente de sua saída.

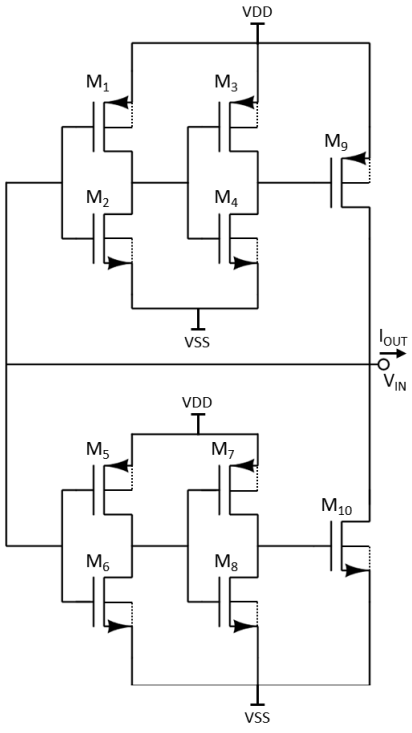


FIGURA 4.3. Esquemático do circuito grampeador proposto.

O funcionamento do circuito pode ser explicado brevemente da seguinte maneira: os transistores M_1 a M_8 formam inversores, que controlam de forma análoga os transistores M_9 e M_{10}. Se por exemplo, a tensão de entrada for superior a um determinado

valor, definido de acordo com as dimensões adotadas para os dispositivos, as tensões nos terminais de porta de M_9 e M_{10} serão altas o suficiente para fazer M_9 entrar em corte e M_{10} drenar uma corrente elevada a partir do nó de entrada (onde se lê o potencial V_{IN}) para compensar um aumento adicional da tensão. Situação análoga ocorre na condição inversa, ou seja, se a tensão de entrada for inferior a um valor pré-definido, porém, neste caso, a corrente de saída é escoada pelo nó de entrada a partir de M_9. Uma comparação entre as características de saída DC do grampeador original e da versão cascode geradas por simulação é mostrada na FIGURA 4.5, que torna bem perceptível a diferença das inclinações das curvas próximo aos valores convencionados como pixels branco e preto. Vale destacar que esta arquitetura comporta a ampliação do número de inversores em cascata para produzir uma característica com transições ainda mais verticais. Entretanto, versões ampliadas do grampeador apresentam como principal desvantagem o aumento do tempo de resposta. Quando testadas na CNN, as versões do grampeador com um número maior de inversores CMOS que o apresentado na FIGURA 4.4 provocaram instabilidades que invalidaram sua aplicação. Portanto, a configuração da FIGURA 4.4 foi escolhida por seu resultado ótimo.

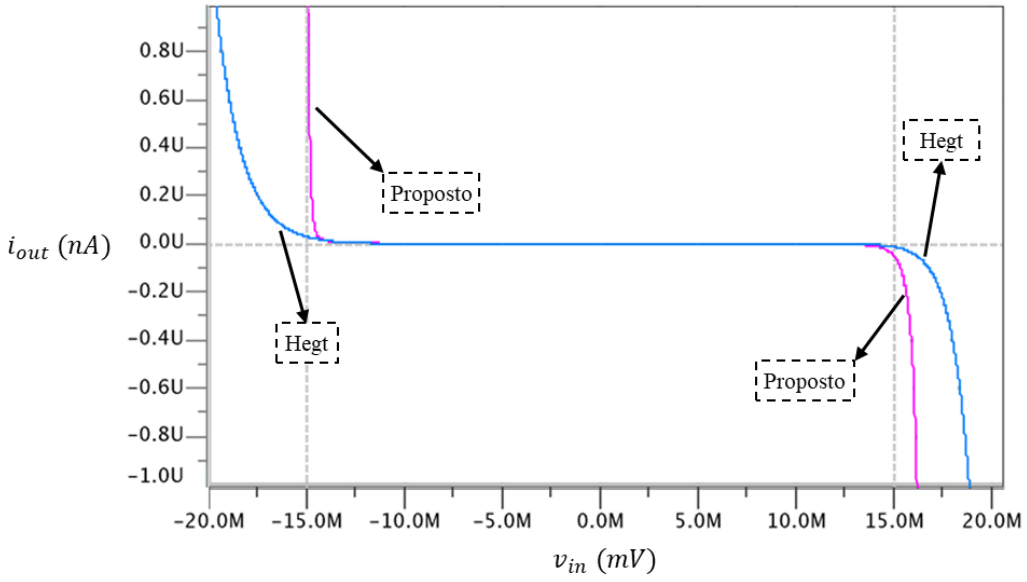


FIGURA 4.4. Curva característica DC da saída dos circuitos grampeadores.

4.1.2 Circuito Completo

O diagrama esquemático do circuito completo da célula da CNN, após os aprimoramentos descritos anteriormente, é exibido na FIGURA 4.6. As modificações em relação à arquitetura original, presente na FIGURA 2.28, estão destacadas.

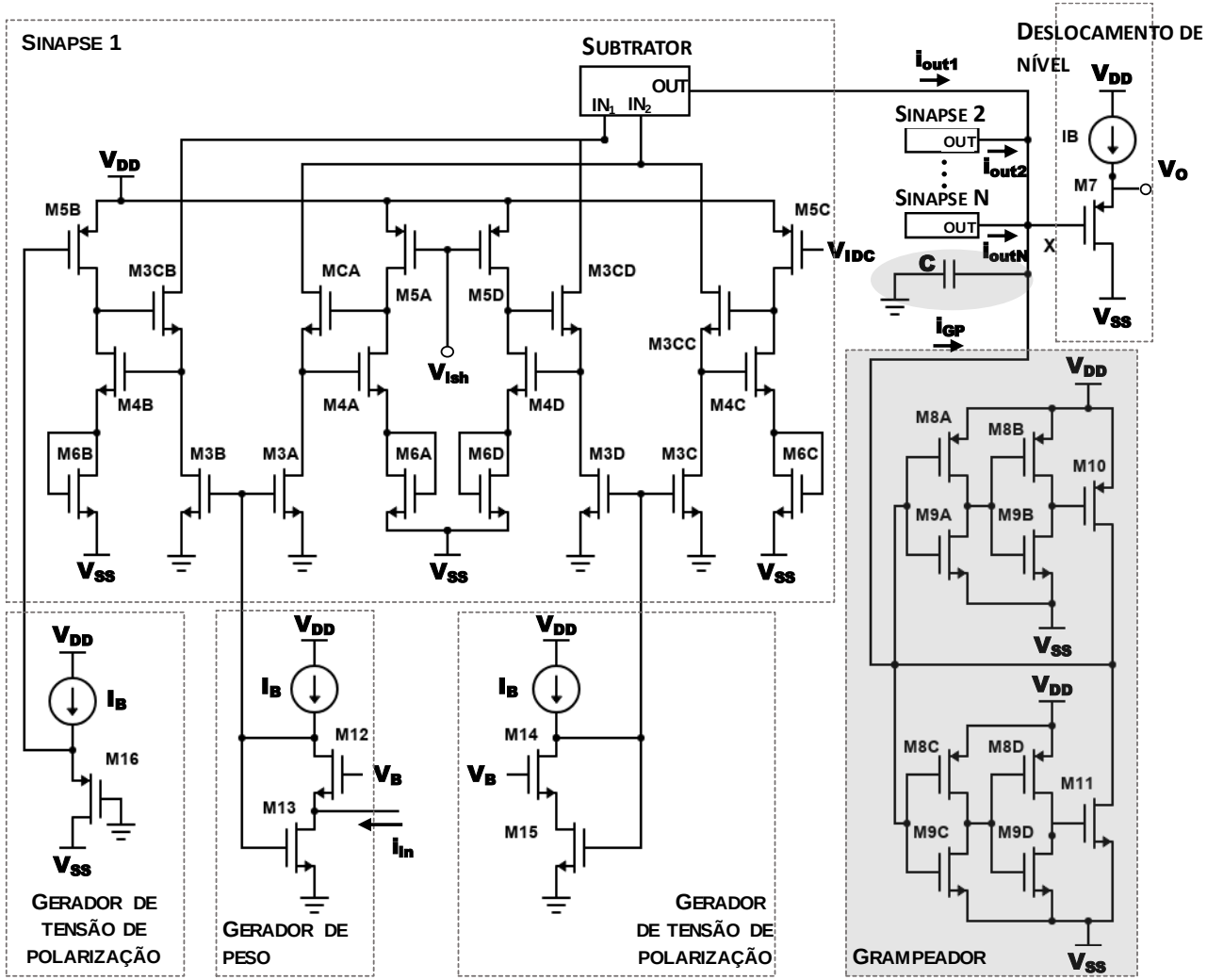


FIGURA 4.5. Circuito da célula da CNN aprimorado.

4.2 RESULTADOS

A CNN foi simulada incorporando as melhorias citadas, em um primeiro momento utilizando coeficientes conhecidos da literatura convertidos para sinais de corrente. Testaram-se então funções bipolares em duas situações: a composição de duas funções tradicionais em sequência, onde cada camada processa uma delas individualmente, e operações de duas camadas, empregando, respectivamente, as configurações 2LFF-CNN e 2LFB-CNN. Em todas elas a CNN respondeu como esperado, gerando imagens sem erros, dado o caráter binário das funções envolvidas.

A terceira categoria de funções simuladas presente no capítulo consiste em operações que trabalham em tons de cinza, representadas aqui pela filtragem de imagens, onde procurou-se explorar múltiplas configurações da 2L-CNN para avaliação de seu desempenho.

4.2.1 Funções Bipolares Tradicionais

4.2.1.1 Combinação de operações A - Dilatação e Detecção de Bordas

Neste caso, foi escolhida uma sequência com as funções desacopladas dilatação e detecção de bordas, descritas na seção 3.3, ocupando, respectivamente, a primeira e a segunda camadas. Portanto, esta combinação primeiramente expande o objeto preto nas direções horizontais e verticais e deixa apenas a borda expandida ao final do processamento. Para tal, a matriz C_2 recebeu os coeficientes que corresponderiam à matriz B da segunda operação em um caso tradicional, sendo executada sobre a saída da primeira camada. Devido a erros encontrados inicialmente em alguns pixels gerados pela operação de detecção de bordas nessa situação, foi efetuada uma alteração do valor do coeficiente central da matriz A_2 para zero, o que reduz a sua velocidade, mas não altera o resultado teórico da função. Após essa mudança, o funcionamento da CNN passou a ser o esperado. As FIGURAS 4.7 e 4.8 mostram, respectivamente, os coeficientes envolvidos e um exemplo de simulação.

A_1		
Iz	Iz	Iz
Iz	75	Iz
Iz	Iz	Iz

B_1		
Iz	75	Iz
75	75	75
Iz	75	Iz

C_1		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

D_1
337,5

A_2		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

B_2		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

C_2		
-34,13	-34,13	-34,13
-34,13	350	-34,13
-34,13	-34,13	-34,13

D_2
-34,13

Sinais de Fronteira	
u_{b1}	-15 mV
y_{b1}	-15 mV
u_{b2}	0
y_{b2}	0

Aplicação de Imagens				
Função	Entrada		Saída	
Dilatação	Im_{E1}	X_{O1}	Im_{S1}	Y_1
Det. Borda	Im_{E2}	Y_1	Im_{S2}	Y_2

FIGURA 4.6. Coeficientes da combinação de operações A. Os coeficientes são dados em nA. $I_Z = 1.14\ nA$ é o valor de correção do offset dos multiplicadores.

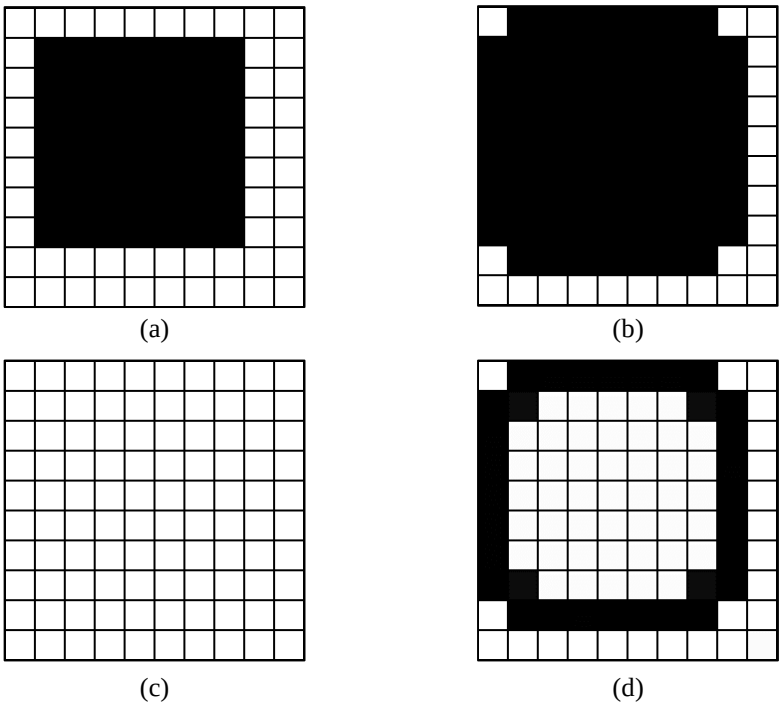


FIGURA 4.7. Resultados de simulação da combinação de operações A. (a) X_{01}. (b) Y_1. (c) X_{02}. (d) Y_2. Tempo de Resposta: 60,5 µs.

4.2.1.2 Combinação de operações B - Cobertura e Diferença Lógica

Este exemplo é composto pela operação cobertura, explicada na seção 3.3, na primeira camada e a operação diferença lógica na segunda camada. Esta última funciona individualmente em cada pixel e pode ser descrita da seguinte forma: aplicando-se duas imagens binárias Im_1 (entrada) e Im_2 (estado inicial), a resposta será uma imagem binária contendo o complemento lógico de Im_1 relativo a Im_2, ou seja, a imagem na saída da rede traz os elementos de cor preta existentes em Im_2 que são brancos em Im_1. A diferença foi usada aqui para comparar o resultado da cobertura com a imagem original, logo, o resultado desta composição é uma imagem contendo apenas os pixels pretos que foram produzidos pela cobertura. Para adaptá-la à segunda camada, a matriz C_2 recebeu os coeficientes que pertencem à matriz A. Os coeficientes podem ser visualizados na FIGURA 4.9, enquanto os resultados de um exemplo de simulação encontram-se na FIGURA 4.10.

A_1		
62,5	62,5	62,5
62,5	250	62,5
62,5	62,5	62,5

B_1		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

C_1		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

D_1
312,5

A_2		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

B_2		
Iz	Iz	Iz
Iz	-136,5	Iz
Iz	Iz	Iz

C_2		
Iz	Iz	Iz
Iz	150	Iz
Iz	Iz	Iz

D_2
-136,5

Sinais de Fronteira	
u_{b1}	-15 mV
y_{b1}	-15 mV
u_{b2}	0
y_{b2}	0

Aplicação de Imagens				
Função	Entrada		Saída	
Cobertura	Im_{E1}	X_{O1}	Im_{S1}	Y_1
Diferença Lógica	Im_{E2}	Y_1	Im_{S2}	Y_2
	Im_{E1}	U_2		

FIGURA 4.8. Coeficientes da combinação de operações B. Os coeficientes são dados em nA. $I_Z = 1.14\ nA$ é o valor de correção do offset dos multiplicadores.

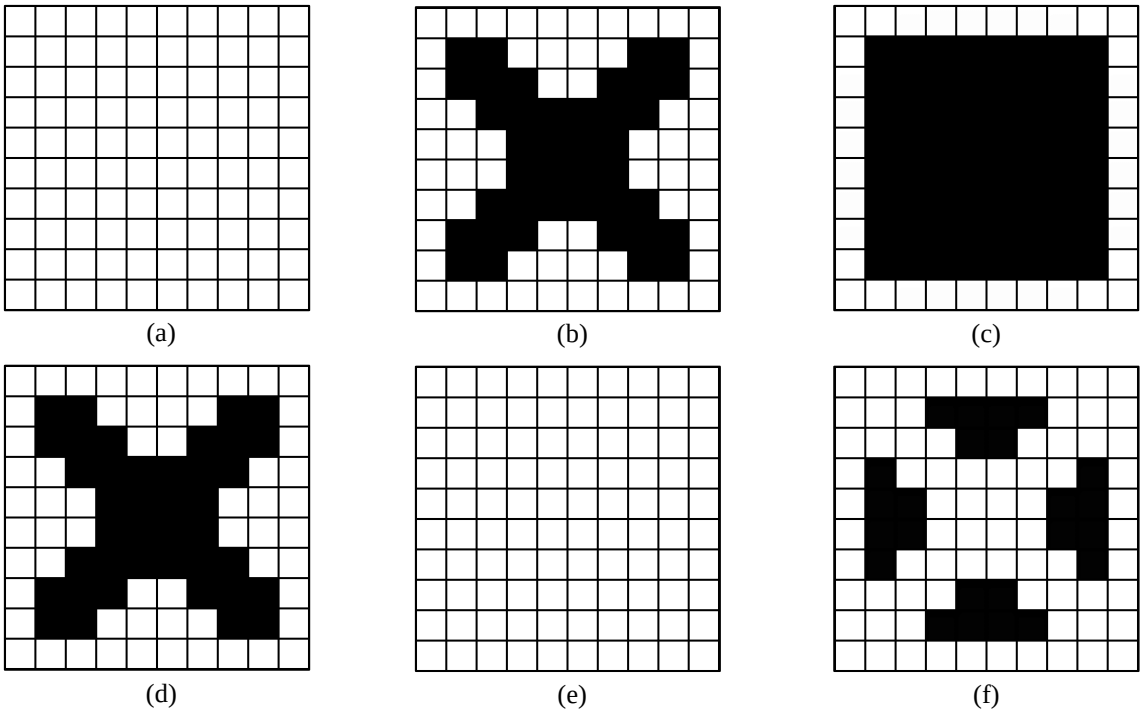


FIGURA 4.9. Resultados de simulação da combinação de operações B. (a) U_1. (b) X_{01}. (c) Y_1. (d) U_2. (e) X_{02}. (f) Y_2. Tempo de Resposta: 27 µs.

4.2.2 Funções Bipolares de Duas Camadas

4.2.2.1 Detecção de linha central

A operação de detecção de linha central, descrita em (YANG, NISHIO e USHIDA, 2003), ilustra o principal mecanismo utilizado na maioria das funções que utilizam ambas as camadas e consiste na utilização destas camadas para realizar uma sequência de eliminações de pixels pretos, a partir dos extremos, até ser atingida uma determinada condição. Para o caso específico de detecção de linha central, a remoção ocorre em uma dimensão nos extremos do objeto, sendo realizada pelas camadas de forma alternada, como mostra a FIGURA 4.11. Seu *template*, mostrada na FIGURA 4.12 para uma direção horizontal de detecção pode ser aproveitada também para a direção vertical após ser feita uma rotação de 90 graus nas matrizes. A FIGURA 4.13 ilustra os resultados de uma simulação.

FIGURA 4.10. Exemplo do princípio de funcionamento da função detecção de linha central. (a)-(f) evolução temporal da resposta. Traduzida de (YANG, NISHIO e USHIDA, 2003).

A_1		
Iz	Iz	Iz
Iz	200	Iz
Iz	Iz	Iz

B_1		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

C_1		
Iz	Iz	Iz
50	100	50
Iz	Iz	Iz

D_1
-227,5

A_2		
Iz	Iz	Iz
Iz	200	Iz
Iz	Iz	Iz

B_2		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

C_2		
Iz	Iz	Iz
-45,5	100	-45,5
Iz	Iz	Iz

D_2
-45,5

Sinais de Fronteira	
u_b	0
y_b	0

Aplicação de Imagens			
Entrada		Saída	
Im_E	X_{O1} e X_{O2}	Im_S	Y_2

FIGURA 4.11. Coeficientes da função detecção de linha central. Os coeficientes são dados em nA. $I_Z = 1.14\ nA$ é o valor de correção do offset dos multiplicadores.

4.2.2.2 Detecção de ponto central

Em algumas aplicações com imagens, a detecção do ponto central de um objeto é uma etapa bem importante, pois fornece uma posição de referência do mesmo. Devido às possíveis variações do formato dos objetos a definição de ponto central pode ser ambígua. Neste trabalho ele é tratado como o ponto cujas coordenadas são os valores médios das coordenadas dos extremos do objeto nas direções horizontal e vertical. Nesse caso, o processamento é feito em cima de blocos retangulares que representam os objetos, o que pode ser obtido a partir do uso da função sombreamento, atuando simultaneamente nas

duas dimensões, conforme explicado em (YANG, NISHIO e USHIDA, 2003). As FIGURAS 4.14 e 4.15 apresentam os coeficientes correspondentes e um exemplo de resultado de simulação, respectivamente.

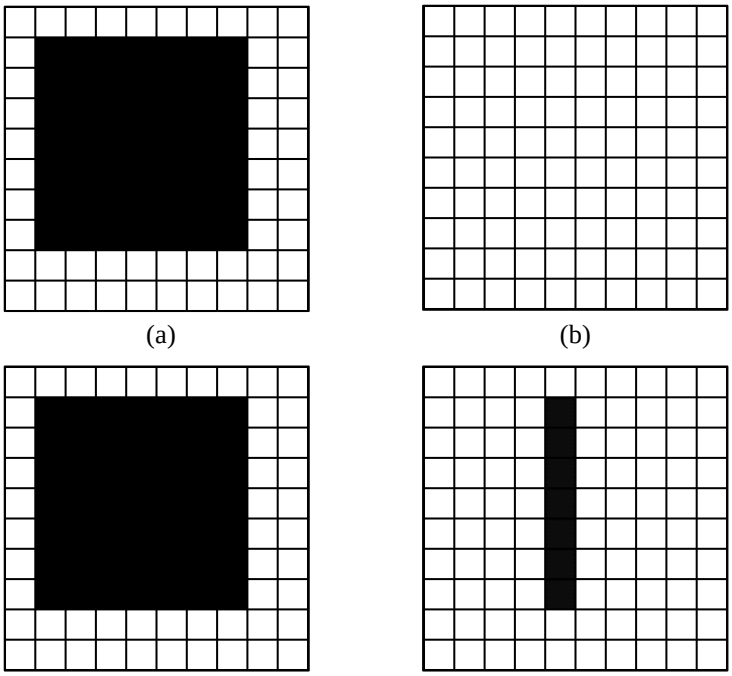


FIGURA 4.12. Resultados de simulação da função detecção de linha central. (a) X_{01}. (b) Y_1. (c) X_{02}. (d) Y_2. Tempo de Resposta: 63,5 µs.

A_1		
Iz	Iz	Iz
Iz	100	Iz
Iz	Iz	Iz

B_1		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

C_1		
Iz	50	Iz
50	100	50
Iz	50	Iz

D_1
-318,5

A_2		
Iz	Iz	Iz
Iz	100	Iz
Iz	Iz	Iz

B_2		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

C_2		
Iz	-45,5	Iz
-45,5	100	-45,5
Iz	-45,5	Iz

D_2
-45,5

Sinais de Fronteira	
u_b	0
y_b	0

Aplicação de Imagens			
Entrada		Saída	
Im_E	X_{O1} e X_{O2}	Im_S	Y_2

FIGURA 4.13. Coeficientes da função detecção de ponto central. Os coeficientes são dados em nA. $I_Z = 1.14\ nA$ é o valor de correção do offset dos multiplicadores.

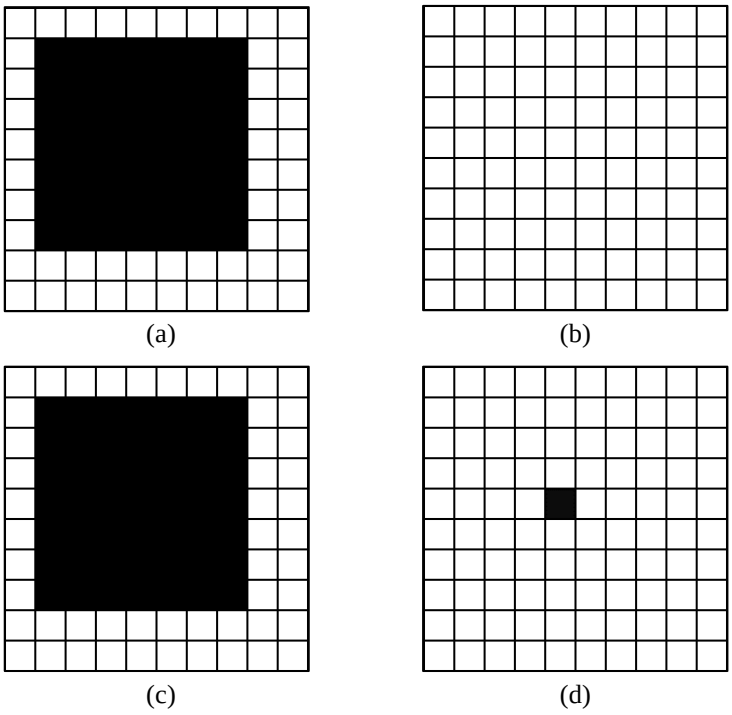


FIGURA 4.14. Resultados de simulação da função detecção de ponto central. (a) X_{01}. (b) Y_1. (c) X_{02}. (d) Y_2. Tempo de Resposta: 84,5 µs.

4.2.2.3 Divisão de objetos pela metade

A função de divisão pela metade é demonstrada em (YANG, NISHIO e USHIDA, 2003), separando objetos em duas partes que possuem a mesma área. A ideia dessa operação é realizar a remoção horizontal dos pixels pretos pela esquerda em uma camada e pela direita na outra, até que se chegue ao pixel central. Ao término do processamento, a saída de cada camada terá uma das partes de cada objeto. As FIGURAS 4.16 e 4.17, respectivamente, mostram os coeficientes correspondentes e um exemplo aplicado.

4.2.2.4 Separação de objetos

A operação de separação de objetos extrai objetos marcados de uma imagem, reproduzindo-os de forma isolada dos demais (YANG, NISHIO e USHIDA, 2003). Isso resulta em duas imagens em que uma delas será composta dos objetos marcados e a outra possuirá o restante, e pode ser útil para extração de caracteres em um texto, por exemplo. Para o funcionamento adequado, deve-se fornecer além da imagem a ser processada, uma segunda imagem com pixels pretos para servir como marcadores. Em consequência, os objetos marcados serão aqueles em que pelo menos um marcador possua a mesma

coordenada de uma de suas partes. Os coeficientes que executam tal procedimento podem ser visualizados na FIGURA 4.18. Já a FIGURA 4.19 contém resultados de uma simulação da função.

A_1		
Iz	Iz	Iz
50	100	-45,5
Iz	Iz	Iz

B_1		
Iz	Iz	Iz
Iz	50	50
Iz	Iz	Iz

C_1		
Iz	Iz	Iz
50	Iz	-45,5
Iz	Iz	Iz

D_1
-91

A_2		
Iz	Iz	Iz
-45,5	100	50
Iz	Iz	Iz

B_2		
Iz	Iz	Iz
50	50	Iz
Iz	Iz	Iz

C_2		
Iz	Iz	Iz
-45,5	Iz	50
Iz	Iz	Iz

D_2
-91

Sinais de Fronteira	
u_b	0
y_b	0

Aplicação de Imagens			
Entrada		Saída	
Im_E	U_1, X_{O1}, U_2 e X_{O2}	Im_{S1}	Y_1
-	-	Im_{S2}	Y_2

FIGURA 4.15. Coeficientes da função divisão pela metade de objetos. Os coeficientes são dados em nA. $I_Z = 1.14\ nA$ é o valor de correção do offset dos multiplicadores.

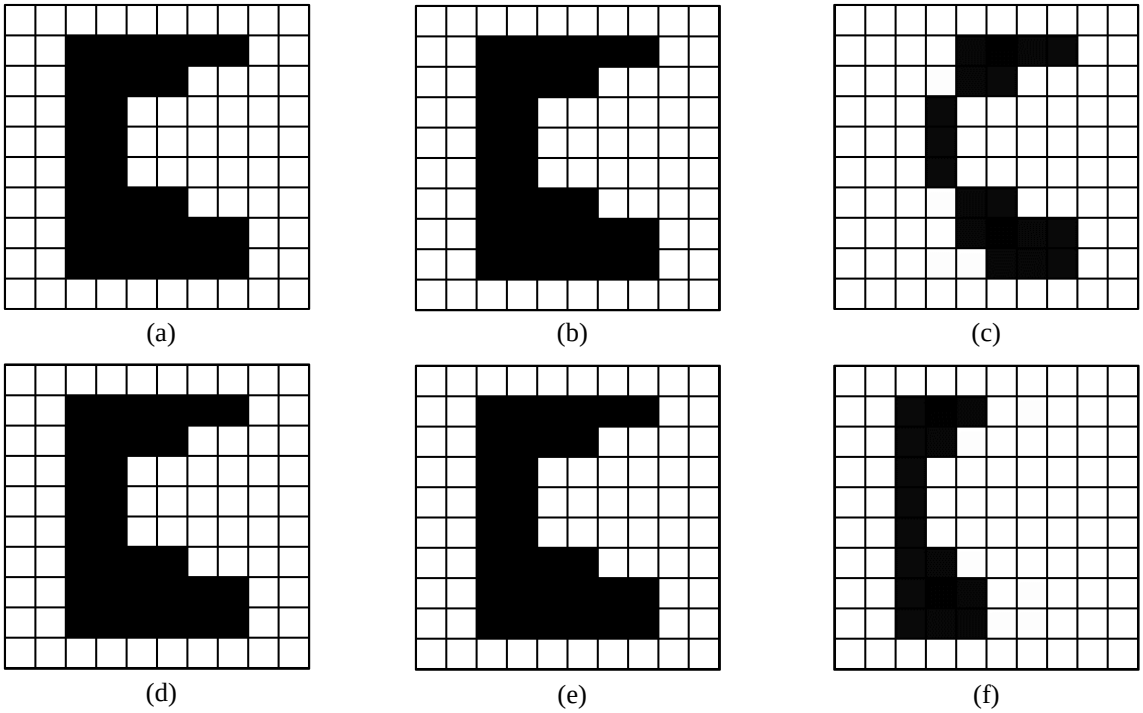


FIGURA 4.16. Resultados de simulação da função divisão pela metade de objetos. (a) U_1. (b) X_{01}. (c) Y_1. (d) U_2. (e) X_{02}. (f) Y_2. Tempo de Resposta: 106,5 µs.

A_1		
25	25	25
25	100	25
25	25	25

B_1		
Iz	Iz	Iz
Iz	200	Iz
Iz	Iz	Iz

C_1		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

D_1
12,5

A_2		
Iz	Iz	Iz
Iz	100	Iz
Iz	Iz	Iz

B_2		
Iz	Iz	Iz
Iz	Iz	Iz
Iz	Iz	Iz

C_2		
Iz	Iz	Iz
Iz	-45,5	Iz
Iz	Iz	Iz

D_2
-45,5

Sinais de Fronteira	
u_b	0
y_b	0

Aplicação de Imagens			
Entrada		Saída	
Im_E	U_1 e X_{O2}	Im_{S1}	Y_1
Marcadores	X_{O1}	Im_{S2}	Y_2

FIGURA 4.17. Coeficientes da função separação de objetos. Os coeficientes são dados em nA. $I_Z = 1.14\ nA$ é o valor de correção do offset dos multiplicadores.

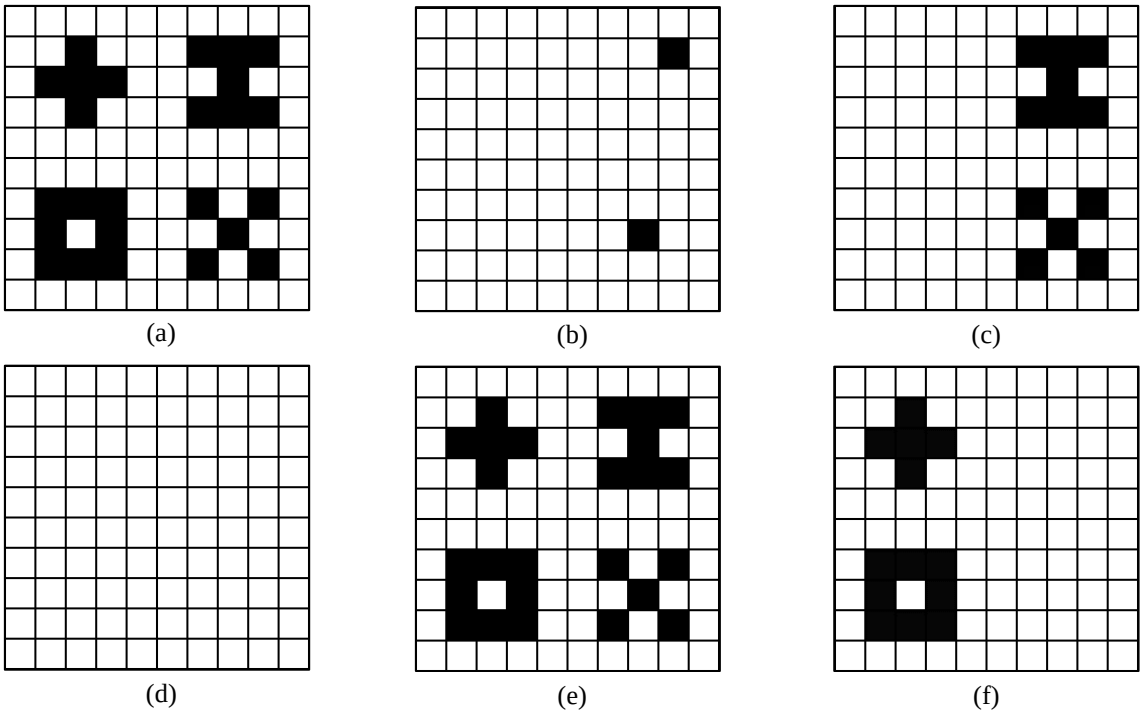


FIGURA 4.18. Resultados de simulação da função separação de objetos. (a) U_1. (b) X_{01}. (c) Y_1. (d) U_2. (e) X_{02}. (f) Y_2. Tempo de Resposta: 34,5 µs.

4.2.3 Filtros de Imagens

Como já citado na seção 3.2.2, os filtros considerados foram projetados utilizando-se o método da função de transferência no domínio da frequência, com os exemplos contidos nesta seção tendo sido objetos de análise em (ANDRADE, SANTANA, *et al.*, 2020). Os coeficientes utilizados para aplicação na rede foram obtidos a partir de múltiplas tentativas de treinamento, escolhendo-se os melhores candidatos encontrados para cada filtro. A TABELA 4.1 resume os seus valores, já convertidos em correntes.

Os coeficientes dos filtros foram então empregados nas simulações do circuito da 2LCNN, utilizando cinco imagens de teste para cada caso, entre as quais encontram-se aquelas representadas na FIGURA 4.20. As respostas obtidas foram então comparadas quantitativamente com a filtragem teórica, baseada na aplicação da FFT, e com os resultados do modelo computacional ideal da rede. Os erros relativos individuais dos pixels foram computados considerando as cinco imagens aplicadas em cada filtro, compondo os histogramas das FIGURAS 4.21 e 4.22. E_{SM} e E_{SF} são os erros relativos da resposta do circuito em relação ao modelo ideal e à filtragem teórica, respectivamente. Além do levantamento dos erros individuais, as raízes dos seus valores quadráticos médios (RMS) foram incluídas, e os erros entre o modelo e o filtro teórico, E_{MF}, também são representados pelo seu valor RMS. Um resumo destas métricas é retratado pela TABELA 4.2.

TABELA 4.1. Especificações e coeficientes dos filtros. PB: passa-baixas; PA: passa-altas.

Filtros (Butterworth IIR, $n = 1$)			Config.	Coeficientes da CNN (nA)														
Nº	Tipo	D_0 (cyc/px)		a_{1_x}	a_{1_y}	a_{1_z}	b_{1_x}	b_{1_y}	b_{1_z}	c_{1_x}	c_{1_y}	c_{1_z}	a_{2_x}	a_{2_y}	a_{2_z}	c_{2_x}	c_{2_y}	c_{2_z}
I	PB	2	2LFF-CNN	12,50	73,00	-16,50	-77,00	-153,00	-74,50	-	-	-	16,00	14,50	-190,50	-12,50	2,00	3,50
II		2	2LFB-CNN	68,25	-42,75	-189,00	-6,00	-165,75	-31,50	78,00	-45,00	23,25	-15,75	51,00	-174,75	-6,75	-2,25	13,50
III	PA	8	2LFF-CNN	18,13	0,00	90,63	-21,88	-6,88	29,38	-	-	-	-5,63	1,88	-200,00	-3,13	-18,75	121,25
IV		8	2LFB-CNN	-16,50	27,75	-114,75	-21,00	-22,50	195,75	20,25	4,50	-95,25	-9,75	-24,75	-110,25	2,25	-6,75	81,75

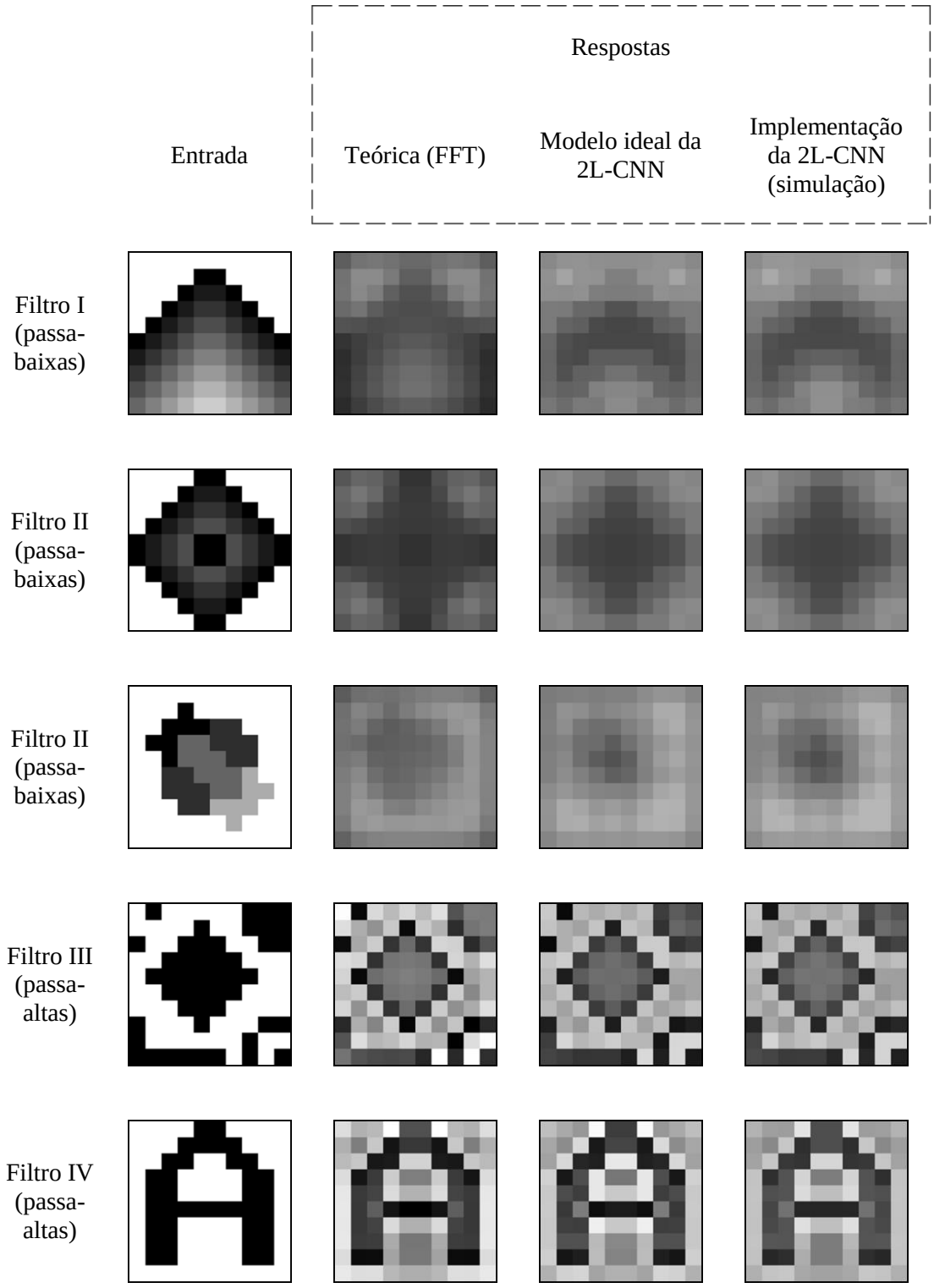


FIGURA 4.19. Exemplo comparativo das respostas dos filtros.

Analis[illegible]s de [illegible]iram [illegible]enor[illegible]is, uma primeira [illegible]leva[illegible]cia pro[illegible]e a resposta do [illegible]escr[illegible]lo, a [illegible]da arquitetura desenvolvida para a 2L-CNN em funções com escala de cinza. Ademais,

embora tanto E_{SF} quanto E_{MF} também exibam graus aceitáveis, seus valores mais elevados indicam que a principal contribuição para as imprecisões encontradas na imagem de saída do circuito advém de limitações da 2L-CNN. Esta percepção é reforçada visualmente pelas imagens da FIGURA 4.20.

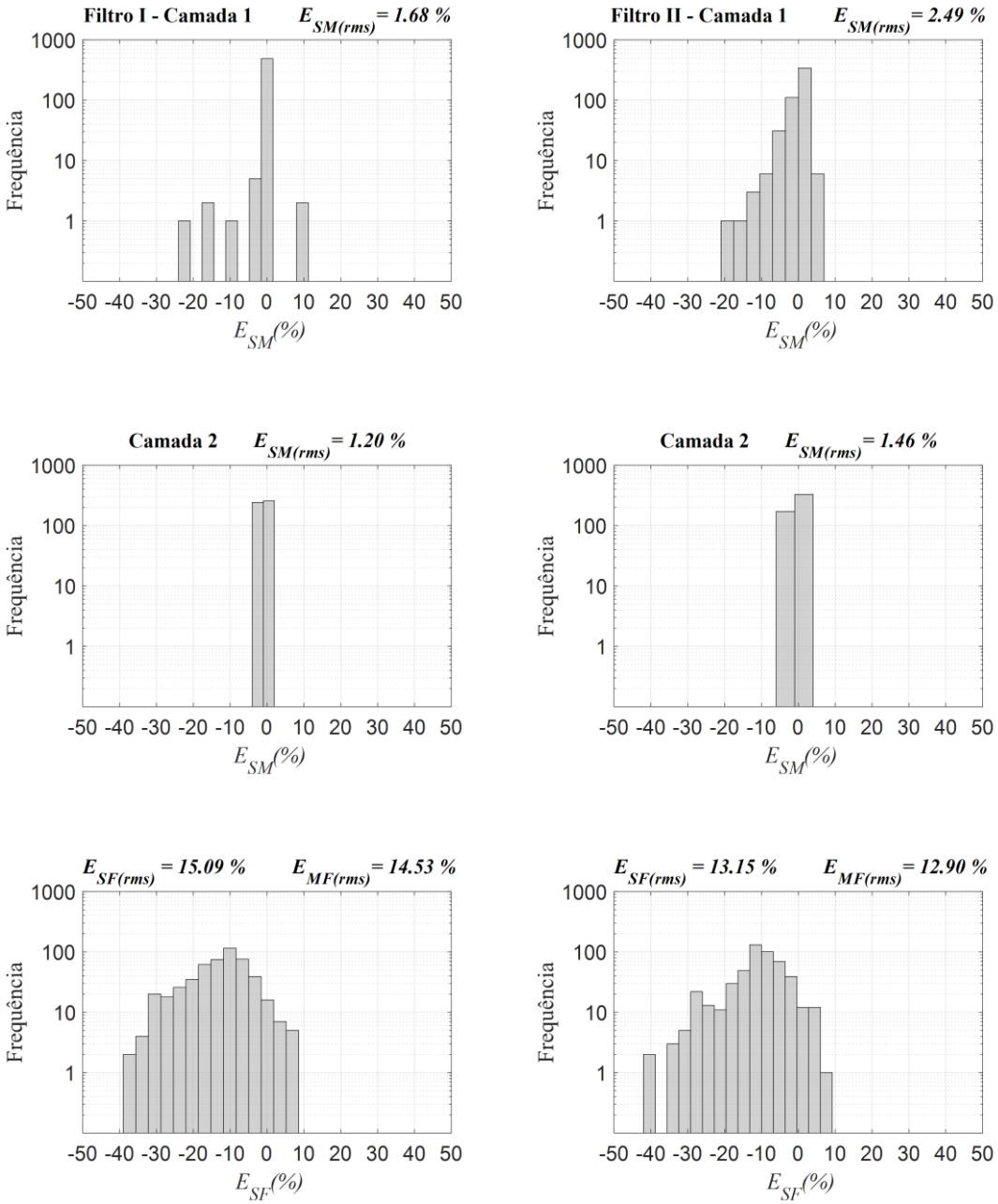


FIGURA 4.20. Distribuição de erro das respostas do circuito da 2L-CNN para os filtros espaciais passa-baixas Butterworth de 1ª ordem. E_{SM}: erro percentual entre os resultados simulados e a resposta do modelo ideal; E_{SF}: erro percentual entre os resultados simulados e a teoria (FFT); E_{MF}: erro percentual entre os resultados do modelo ideal e a teoria (FFT). Filtro I: 2LFF-CNN; Filtro II: 2LFB-CNN.

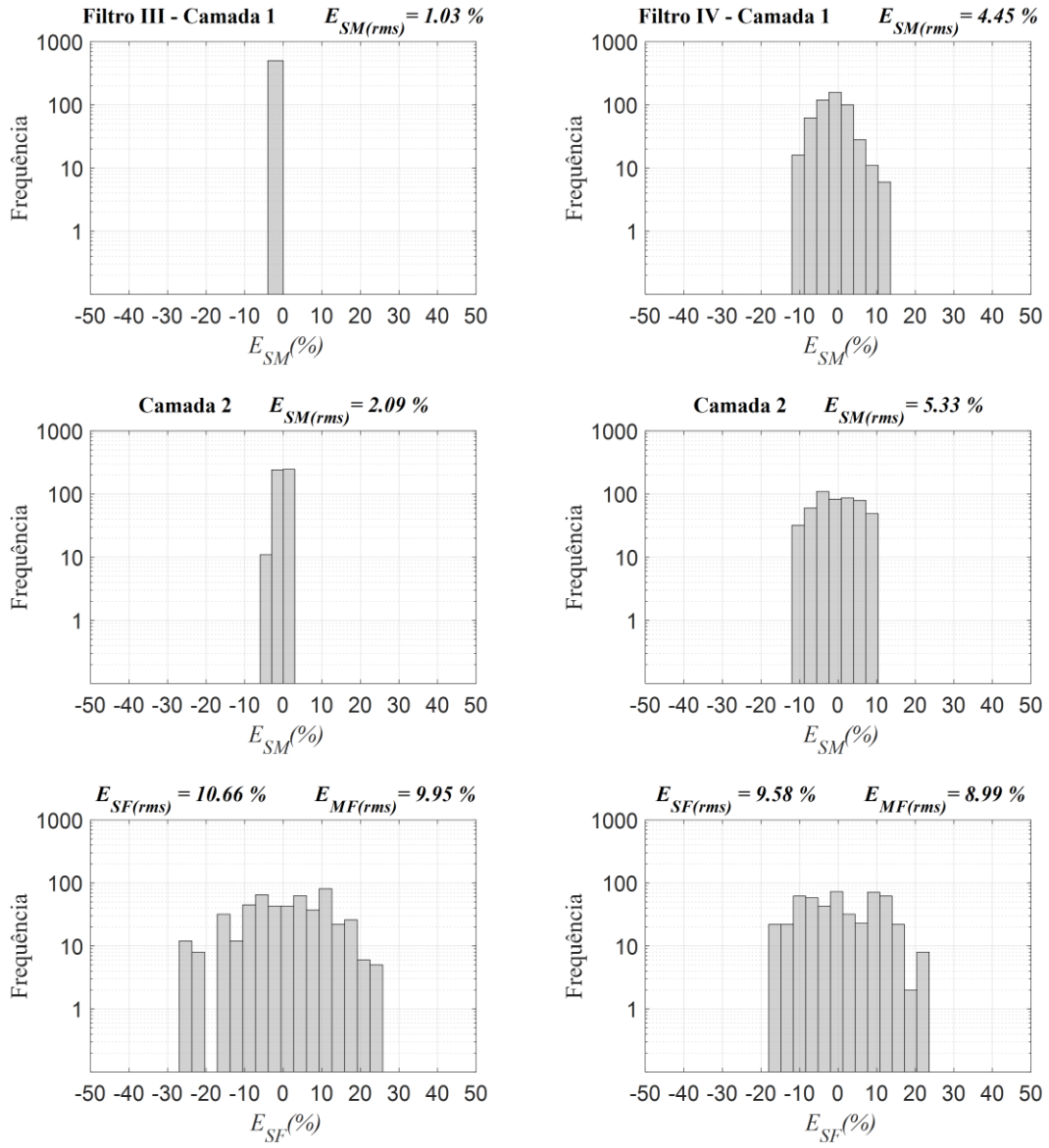


FIGURA 4.21. Distribuição de erro das respostas do circuito da 2L-CNN para os filtros espaciais passa-altas do tipo Butterworth de 1ª ordem. E_{SM}: erro percentual entre os resultados simulados e a resposta do modelo ideal; E_{SF}: erro percentual entre os resultados simulados e a teoria (FFT); E_{MF}: erro percentual entre os resultados do modelo ideal e a teoria (FFT). Filtro III: 2LFF-CNN; Filtro IV: 2LFB-CNN.

TABELA 4.2. Valores RMS dos erros.

	E_{SM} (%)		E_{SF}	E_{MF}
	1ª Camada	2ª Camada		
Filtro I	1,68	1,20	15,09	14,53
Filtro II	2,49	1,46	13,15	12,90
Filtro III	1,03	2,09	10,66	9,95
Filtro IV	4,45	5,33	9,58	8,99

Nota-se também que os filtros II e IV, envolvendo acoplamento mútuo, produziram respostas com níveis de E_{SM} um pouco menores do que seus correlatos desacoplados (I e III), revelando uma vantagem na sua capacidade de filtragem. Adicionalmente, tanto os valores numéricos quanto uma inspeção visual indicam que os testes relacionados aos filtros passa-altas (III e IV) tiveram um desempenho mais próximo do modelo teórico, o que pode estar associado às diferenças na complexidade do espaço de otimização em cada um dos casos. Ademais, enquanto os erros atrelados aos filtros passa-altas demonstram uma clara uniformidade entre as dispersões de cada camada, os histogramas referentes aos filtros passa-baixas indicam que houve uma maior variância na 1ª camada, distinção que possivelmente decorre de diferentes caminhos tomados pelo treinamento no que se refere ao papel de cada camada na dinâmica da operação completa, já que não se procurou controlar diretamente este aspecto.

Uma importante discussão neste contexto envolve a relevância do nível dos erros exibidos nos resultados da CNN para as possíveis aplicações do circuito. Assim, dois pontos são destacados: o primeiro está atrelado às exigências de determinadas aplicações, como é o caso de próteses retinianas, onde o objetivo é a realização de um pré-processamento dos sinais visuais, e que, portanto, demandam uma precisão muito elevada nas operações; ademais, operações associadas pela aplicação de filtros como os tratados nesta seção, como por exemplo a suavização de imagens e o realce de bordas, possui um caráter subjetivo e complexo para ser bem definido de forma quantitativa. Logo, ainda que haja a presença de erros perceptíveis nas comparações das imagens processadas, a resposta da CNN ainda pode ser considerada válida nestas circunstâncias. Um outro objeto de grande relevância neste contexto engloba as incertezas que permeiam o processo de treinamento da CNN e que atuam para dificultar um mapeamento preciso das contribuições de cada etapa necessária para a aplicação da rede nos erros obtidos. Uma implicação deste fato é a dificuldade de se prever o potencial para o aumento do desempenho da filtragem ao se buscar novos coeficientes. Em outras palavras, é possível, mas não garantido, que haja um ou mais conjuntos de parâmetros não encontrados que produzam um resultado significativamente mais próximo do desejado. Por outro lado, o principal fator limitante pode residir na incapacidade intrínseca à estrutura da 2L-CNN para reproduzir exatamente a operação. Desta forma, é razoável supor que a relativa proximidade entre as respostas dos exemplos de 2LFF-CNN e dos exemplos de 2LFB-CNN não representa necessariamente as suas potencialidades e que a partir de

melhoramentos adicionais da metodologia de treinamento as redes com acoplamento mútuo possam ampliar esta vantagem.

5 CONCLUSÃO

A proposta deste trabalho consistiu na realização e treinamento de uma CNN analógica de duas camadas em tecnologia CMOS, usando como base a arquitetura do tipo FSR proposta em (SANTANA, 2013) e dando prosseguimento ao desenvolvimento já tratado em (ANDRADE, 2015a), envolvendo aplicações em processamento de imagens. Essa ampliação da rede visa não apenas a execução de sequências de funções tradicionais como também operações mais complexas do que as que foram demonstradas anteriormente, incluindo também a reprodução de filtros de maior resolução nos cortes.

Nesse contexto, um dos importantes pontos abordados foi o treinamento da CNN, onde além de incorporar novos aprimoramentos no algoritmo que já estava sendo utilizado (CMA), foi também investigada a necessidade da implementação de uma segunda técnica, o GA, que mostrou uma maior capacidade de busca da solução para operações bipolares que apresentam propagação de sinal. A inserção da capacidade do GA trabalhar com uma representação real para os coeficientes da CNN ampliou a flexibilidade disponível para as suas etapas, em especial a codificação e os operadores genéticos, favorecendo a normalização dos pesos e a formulação de exemplares que aproveitam as caraterísticas e a dinâmica da CNN. Um estudo comparativo revelou um destaque particular para a operação de recombinação do cruzamento com distribuição gaussiana ao fornecer ao GA uma maior capacidade de convergência.

Uma terceira abordagem para o treinamento da rede buscando conciliar as vantagens do CMA e do GA culminou em um algoritmo híbrido, originado da combinação das duas técnicas. Neste contexto, testes comparativos com funções bipolares revelaram que esta vertente pode antecipar a convergência do processo, apresentando, contudo, um aumento no tempo médio de processamento em cada geração.

Adicionalmente, a inclusão de novas sinapses para conexão das camadas amplificou o efeito de imperfeições do circuito, prejudicando a resposta do sistema. Por conta de tais limitações, um estudo mais aprofundado do comportamento da rede foi realizado e alguns ajustes foram propostos para melhoria do desempenho, permitindo a execução de funções bipolares de duas camadas em nível de simulação com erros desprezíveis. Por sua vez, a aplicação de coeficientes gerados por treinamento visando a filtragem de imagens, que admite sinais na escala de cinza, revelou uma boa proximidade entre o circuito simulado e o modelo computacional ideal, fornecendo mais um indicativo importante da funcionalidade da arquitetura.

Uma situação mais complexa foi constatada nas operações em tons de cinza testadas, mais especificamente os filtros de imagem. Tendo em vista a sua maior sensibilidade às imperfeições do processamento, bem como a sua exigência de uma elevada precisão dos coeficientes, esforços mais intensos foram necessários tanto para a obtenção dos coeficientes adequados quanto para a análise do desempenho da rede. Apesar destes entraves, a 2L-CNN simulada apresentou um desempenho próximo ao retratado por um modelo computacional ideal, corroborando a viabilidade da arquitetura de célula analógica empregada nas aplicações propostas.

Apesar dos relevantes avanços deste trabalho, muitas oportunidades de evolução são vislumbradas. A metodologia de treinamento, por um lado, fornece muitas possibilidades visando o aumento do desempenho. O GA pode receber a adição de novos tipos de operadores, dotados de mais versatilidade, buscado uma especialização adaptável a certas condições encontradas. Uma mudança da organização dos indivíduos, como a proposta pelo Algoritmo Genético Celular, cujo desempenho no treinamento de CNN já foi abordado de forma incipiente em (OLIVEIRA, 2017), carece ainda de um exame específico mais minucioso. Ademais, o Algoritmo Híbrido exibe uma margem para aceleração a partir de novos aprimoramentos que estendam o paralelismo característico do GA para a etapa referente à busca local. Um terceiro caminho, implicando em um desvio mais drástico, mas não menos relevante, passa pela adoção de outras técnicas de otimização conhecidas, preferencialmente aquelas que compõem o conjunto das meta-heurísticas.

A verificação da aplicabilidade da arquitetura da 2L-CNN também deve ser expandida. Um aprofundamento da análise da resposta da rede no domínio da frequência permite, ao mesmo tempo, ampliar a compreensão das vantagens proporcionadas pela acoplagem mútua e contribuir na consolidação do seu papel na filtragem de imagens. Por outro lado, um enfoque mais voltado para funções que desfrutam de uma proximidade do funcionamento dos sistemas biológicos visuais pode abrir uma perspectiva para o emprego da rede para fins biomédicos. Neste sentido, inclusive, espera-se que o caráter biomórfico do sistema favoreça naturalmente a sua aplicação.

No que concerne ao circuito analógico da 2L-CNN realizado, algumas questões ainda demandam futuros tratamentos. Para uma completa verificação de sua funcionalidade, é necessário avaliar o seu comportamento frente variações de fatores como temperatura, tensão de alimentação e parâmetros do processo de fabricação. Em se

tratando a CNN de um sistema em que a uniformidade dos blocos é capital, a sensibilidade a tais aspectos pode comprometer crucialmente o seu desempenho, e uma análise mais profunda serve de fundamentação para as devidas mitigações. Por outro lado, em consonância com uma perpétua marca do projeto de circuitos microeletrônicos, a arquitetura desenvolvida neste trabalho está sempre passível a aperfeiçoamentos em seus blocos constituintes, não cabendo desprezar até mesmo a sua substituição. Desta constatação decorre que a incorporação de outros exemplares de multiplicadores, como os propostos em (SOUSA, ANDRADE, *et al.*, 2019) e (CARDOSO, SCHNEIDER e SANTANA, 2018) para aplicações com CNN, e de grampeadores, ainda que solicite a adoção de adaptações, pode representar um salto proveitoso na busca para um desempenho mais eficaz.

TRABALHOS PUBLICADOS

ANDRADE, F. S. et al. **CNN Learning for Image Processing: Center of Mass versus Genetic Algorithms.** IEEE 10th Latin American Symposium on Circuits and Systems. Armenia: [s.n.]. 2019.

ANDRADE, F. S. et al. **A CMOS Analog Two-Layer Full Signal Range Cellular Neural Network for Image Filtering**. 2020 33rd Symposium on Integrated Circuits and Systems Design (SBCCI). Campinas: [s.n.]. 2020.

DE SOUSA, A. J. S. et al. **A Very Compact CMOS Analog Multiplier for Application in CNN Synapses.** IEEE 10th Latin American Symposium on Circuits and Systems. Armenia: [s.n.]. 2019.

GONÇALVES, G. C. et al. **Evaluation of Distortion Level in Analog Multipliers through DC Analysis Only.** IEEE 10th Latin American Symposium on Circuits and Systems. Armenia: [s.n.]. 2019.

MEHDIPOUR, E. et al. **Modeling Short-Channel Effects for Design by Hand with MOSFET Series Association.** Latin American Electron Devices Conference. Armenia: [s.n.]. 2019.

DE SOUSA, A. J. S. et al. **CMOS Analog Four-Quadrant Multiplier Free of Voltage Reference Generators.** 32nd Symposium on Integrated Circuits and Systems Design (SBCCI). Sao Paulo: [s.n.]. 2019.

FERNANDES, A. A. et al. **Low Saturation Onset MOS Transistor: an Equivalent Network.** 2019 34th Symposium on Microelectronics Technology and Devices (SBMicro). São Paulo: [s.n.]. 2019.

GONÇALVES, G. C. et al. **Using Two-Dimensional DC Characterization to Improve Distortion Level of Analog Multipliers.** 4th International Symposium on Instrumentation Systems, Circuits and Transducers (INSCIT). São Paulo: [s.n.]. 2019.

DOS SANTOS, E. S. et al. **Improvements on the Design of the Low Saturation Onset Transistor.** 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS). Glasgow: [s.n.]. 2020.

D'EÇA, L. C. et al. **Proportional Source Transconductances Integrator for CMOS Analog Filtering with Calibration.** IEEE International Symposium on Circuits and Systems (ISCAS). Sevilha: [s.n.]. 2020.

DE SOUSA A. J. S. et al. Compact CMOS Analog Multiplier Free of Voltage Reference Generators. **Journal of Integrated Circuits and Systems (JICS),** 15, n. 3, 2020. 1-12.

TRABALHOS SUBMETIDOS

ANDRADE F. S. et al. **A Graphical Interface Learning Tool for Image Processing through Analog CNN** [Periódico], 2020.

GONÇALVES G. C. et al. **On the Distortion Analysis of Electronic Analog Multipliers** [Periódico], 2020.

REFERÊNCIAS BIBLIOGRÁFICAS

AHMAD, F. et al. **Performance comparison of gradient descent and Genetic Algorithm based Artificial Neural Networks training**. 10th International Conference on Intelligent Systems Design and Applications. Cairo: [s.n.]. 2010. p. 604-609.

ANDRADE, F. S. **Filtragem de Imagens em Escala de Cinza por Meio de Rede Neuronal Celular Analógica em Tecnologia CMOS**. Salvador: Dissertação de Mestrado - Universidade Federal da Bahia, 2015a.

ANDRADE, F. S. et al. **Image Filtering in a CMOS Analog CNN**. Proceedings of the 2015 IEEE 6th Latin American Symposium on Circuits and Systems. Montevideo: [s.n.]. 2015b. p. 1-4.

ANDRADE, F. S. et al. **CNN Learning for Image Processing:** Center of Mass versus Genetic Algorithms. IEEE 10th Latin American Symposium on Circuits and Systems. Armenia: [s.n.]. 2019.

ANDRADE, F. S. et al. **A CMOS Analog Two-Layer Full Signal Range Cellular Neural Network for Image Filtering**. 2020 33rd Symposium on Integrated Circuits and Systems Design (SBCCI). Campinas: [s.n.]. 2020.

BELOV, M. P.; ZOLOTOV, O. I. **Optimization of parameters of neural networks by genetic algorithm in the control systems of electromechanical objects**. XVIII International Conference on Soft Computing and Measurements (SCM). St. Petersburg: [s.n.]. 2015. p. 136-138.

BOTOCA, C. **Cellular neural networks assisted automatic detection of elements in microscopic medical images. A preliminary study**. 2014 11th International Symposium on Electronics and Telecommunications (ISETC). Timisoara: [s.n.]. 2014. p. 1-4.

CARDOSO, F. M.; SCHNEIDER, M. C.; SANTANA, E. P. **CMOS analog multiplier with high rejection of power supply ripple**. IEEE 9th Latin American Symposium on Circuits & Systems (LASCAS). Puerto Vallarta: [s.n.]. 2018. p. 1-4.

CHUA, L. O.; ROSKA, T. **Cellular Neural Networks and Visual Computing:** Foundations and Applications. Cambridge: Cambridge University Press, 2002.

CHUA, L. O.; YANG, L. Cellular neural networks: theory. **IEEE Transactions on Circuits and Systems**, 35, n. 10, 1988. 1257-1290.

CUN, Y. L. et al. **Handwritten zip code recognition with multilayer networks**. Proceedings. 10th International Conference on Pattern Recognition. Atlantic City: [s.n.]. 1990. p. 35-40.

ESPEJO, S. et al. A VLSI-oriented continuous-time CNN model. **International Journal of Circuit Theory and Applications**, 24, n. 3, 1996. 341-356.

FERNANDES, A. A. et al. **Low Saturation Onset MOS Transistor:** an Equivalent Network. 2019 34th Symposium on Microelectronics Technology and Devices (SBMicro). São Paulo: [s.n.]. 2019.

GOLDBERG, D. Real-coded Genetic Algorithms, Virtual Alphabets, and Blocking. **Complex Systems**, 1991. 139-167.

GOLDBERG, D. **Genetic Algorithms in Search, Optimization, and Machine Learning**. [S.l.]: Dorling Kindersley Pvt Ltd, 2008.

GOLLISCH, T.; MEISTER, M. L. Eye Smarter Than Scientists Believe: Neural Computations in Circuits of the Retina. **Neuron**, 65, 2010. 150-164.

GONZALEZ, R. C.; WOODS, R. E. **Digital Image Processing**. [S.l.]: Pearson Prentice Hall, v. 3, 2007.

HAO, D.; JI, L.; ZHOU, L. **Rapid vehicle edge detection based on cellular neural network**. 2014 10th International Conference on Natural Computation (ICNC). Xiamen: [s.n.]. 2014. p. 118-122.

HEGT, J. A.; LEENAERTS, D. M. W.; WILMANS, R. T. **A novel compact architecture for a programmable full-range cnn in 0.5 um cmos technology**. Proceedings of Fifth IEEE International Workshop on Cellular Neural Networks and Their Applications. London: [s.n.]. 1998. p. 288–293.

KAMEI, T.; MIZOGUCH, M. **Image Filter Design for Fingerprint Enhancement**. Proceedings of International Symposium on Computer Vision. [S.l.]: [s.n.]. 1995. p. 109-114.

KOZEK, T.; ROSKA, T.; CHUA, L. O. Genetic Algorithm for CNN Template Learning. **IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications**, 40, n. 6, 1993. 392-402.

LAI, J.; WU, P. C. Architectural Design and Analysis of Learnable Self-Feedback Ratio-Memory Cellular Nonlinear Network (SRMCNN) for Nanoelectronic Systems. **IEEE Transactions on Very Large Scale Integration (VLSI) Systems**, 12, n. 11, 2004. 1182-1191.

MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. [S.l.]: Springer, 1996.

MIRZAI, B.; CHENG, Z.; MOSCHYTZ, G. S. **Learning Algorithms for Cellular Neural Networks**. Proceedings of the 1998 IEEE International Symposium on Circuits and Systems. Monterey: [s.n.]. 1998.

MORENO-ARMENDARIZ, M. A.; EGIDIO PAZIENZA, G.; YU, W. Training Cellular Neural Networks with Stable Learning Algorithm. **Advances in Neural Networks - ISNN**, Berlin, 3971, 2006. 558-563.

NIU, S. et al. **Research on Human Retinal Cell Image Restoration**. Proceedings of the 3rd International Congress on Image and Signal. [S.l.]: [s.n.]. 2010.

NOSSEK, J. A. **Design and Learning with Cellular Neural Networks**. Proceedings of the Third IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA-94). Rome: [s.n.]. 1994.

OLIVEIRA, N. C. **Análise de Desempenho do Algoritmo Genético Celular no treinamento de uma CNN**. Salvador: Trabalho de Conclusão de Curso - Universidade Federal da Bahia, 2017.

PLEBE, A.; GALLO, G. **Filtering Echocardiographic Image Sequences in Frequency Domain**. Proceedings of the 2nd International Symposium on Image and Signal Processing and Analysis. [S.l.]: [s.n.]. 2001. p. 238-243.

RODRÍGUEZ-VÁZQUEZ, A. E. A. Current-mode techniques for the implementation of continuous and discrete time cellular neural networks. **IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing**, 40, n. 3, 1993. 132-146.

SANTANA, E. P. **Circuitos Analógicos em Tecnologia CMOS para Implementação de Próteses Retinianas**. Salvador: Tese de Doutorado - Universidade Federal da Bahia, 2013.

SANTANA, E. P.; FREIRE, R. C.; CUNHA, A. I. A. A compact low-power CMOS analog FSR model-based CNN. **Journal of Integrated Circuits and Systems**, 7, n. 1, 2012.

SOUSA, A. J. S. D. et al. **CMOS Analog Four-Quadrant Multiplier Free of Voltage Reference Generators**. 32nd Symposium on Integrated Circuits and Systems Design (SBCCI). São Paulo: [s.n.]. 2019. p. 1-6.

TANAKA, M. et al. **Leaning theory of Cellular Neural Networks based on covariance structural analysis**. 12th International Workshop on Cellular Nanoscale Networks and their Applications (CNNA). Berkeley: [s.n.]. 2010. p. 1-4.

TAVSANOGLU, V.; SAATCI, E. **Feature extraction for character recognition using Gabor-type filters implemented by cellular neural networks**. Proceedings of the 2000 6th IEEE International Workshop on Cellular Neural Networks and their Applications (CNNA). Catania: [s.n.]. 2000. p. 63-68.

VADDI, R. et al. **Cellular neural network based pre-processing for localization of non standard licence plate**. 2011 3rd International Conference on Electronics Computer Technology. Kanyakumari: [s.n.]. 2011. p. 407-411.

YANG, T. **Handbook of CNN Image Processing:** All You Need to Know about Cellular Neural Networks. [S.l.]: Yang's Scientific Research Institute LLC, v. 1, 2002.

YANG, Z. H.; NISHIO, Y.; USHIDA, A. Image processing of two-layer CNNs - Applications and their stability. **IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences**, E85A, n. 9, 2002. 2052-2060.

YANG, Z. H.; NISHIO, Y.; USHIDA, A. Characteristic of mutually coupled two-layer CNN and its stability. **Journal of Circuits, Systems, and Computers**, 12, n. 4, 2003. 473-490.

ZAGHLOUL, K. A. **A Silicon Implementation of a Novel Model for Retinal Processing**. Pennsylvania: Thesis (PhD) - University of Pennsylvania, 2009.

ZHANG, H.; DONG, Z.; XU, G. **Neural networks adaptive control of aircraft engine based on genetic algorithm**. The 26th Chinese Control and Decision Conference (CCDC). Changsha,: [s.n.]. 2014. p. 3518-3522.