



LIPNET COM OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

Trabalho de Conclusão de Curso
Engenharia da Computação

Alessandra Maranhão Soares

Orientador: Prof. Dr. Carmelo José Albanez Bastos Filho

Co-Orientador: Prof. Dr. Bruno José Torres Fernandes



Alessandra Maranhão Soares

LIPNET COM OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS

Monografia apresentada como requisito parcial para
obtenção do diploma de Bacharel em Engenharia de
Computação pela Escola Politécnica de Pernambuco -
Universidade de Pernambuco

Universidade de Pernambuco
Escola Politécnica de Pernambuco
Graduação em Engenharia de Computação

Orientador: Prof. Dr. Carmelo José Albanez Bastos Filho
Coorientador: Prof. Dr. Bruno José Torres Fernandes

Recife - PE, Brasil
4 de dezembro de 2013

MONOGRAFIA DE FINAL DE CURSO

Avaliação Final (para o presidente da banca)*

No dia 13 de 12 de 2013, às 10:00 horas, reuniu-se para deliberar a defesa da monografia de conclusão de curso do discente ALESSANDRA MARANHÃO SOARES, orientado pelo professor Carmelo José Albanez Bastos Filho, sob título LIPNET COM OTIMIZAÇÃO POR ENXAME DE PARTÍCULAS, a banca composta pelos professores:

Sérgio Campello Oliveira

Carmelo José Albanez Bastos Filho

Após a apresentação da monografia e discussão entre os membros da Banca, a mesma foi considerada:

Aprovada Aprovada com Restrições* Reprovada

e foi-lhe atribuída nota: 10,0 (10)

*(Obrigatório o preenchimento do campo abaixo com comentários para o autor)

O discente terá 07 dias para entrega da versão final da monografia a contar da data deste documento.

SÉRGIO CAMPELLO OLIVEIRA

CARMELO JOSÉ ALBANEZ BASTOS FILHO

*Este trabalho é dedicado à minha família que,
sempre me deu todo o suporte necessário para realizar
meus sonhos.*

Agradecimentos

Agradeço, primeiramente, a Deus por permitir que tudo isso se concretizasse. Sem a Sua vontade não seria possível chegar ao fim do curso de graduação.

Agradeço também à minha família: meus pais, Ribamar Soares e Soraia Cavalcante, e minha irmã Vanessa Soares. Agradeço pela força concedida e desculpo-me pelos momentos de ausência, momentos estes destinados a estudos, à realização de projetos e à elaboração deste trabalho.

Agradeço ao meu namorado, Artur Sivini. Por que não dizer que ele também foi um professor para mim?! Sua ajuda foi imprescindível para a minha formação. Agradeço, sobretudo, pela sua presença ao meu lado desde o primeiro dia de aula.

Agradeço também ao meu orientador Prof. Carmelo Bastos e co-orientador Prof. Bruno Fernandes. Eles que foram responsáveis por guiar o desenvolvimento deste trabalho: sempre auxiliando, corrigindo e aconselhando.

Agradeço, por fim, a todos os que colaboraram para minha formação acadêmica. Todos aqueles que contribuíram para que eu me tornasse uma estudante, uma profissional e uma pessoa melhor.

*"Não vos amoldeis às estruturas deste mundo,
mas transformai-vos pela renovação da mente,
a fim de distinguir qual é a vontade de Deus:
o que é bom, o que Lhe é agradável, o que é perfeito."
(Bíblia Sagrada, Romanos 12, 2)*

Resumo

A LIPNet é uma rede neural piramidal com inibição lateral utilizada na visão computacional para reconhecer padrões. Esta rede é baseada no conceito de campos receptivos e inibitórios presente no sistema visual humano e é formada por duas camadas: a primeira bidimensional, na qual ocorre a extração implícita de características; e a segunda unidimensional, responsável pela classificação do padrão de entrada. Porém, o método de treinamento dessa rede é o *Rprop*, que é susceptível à estagnação em mínimos locais. Para contornar esse problema, este trabalho propôs utilizar técnicas de inteligência de enxames para otimização. Portanto, duas arquiteturas foram definidas: Arquiteturas I e II. A primeira visa otimizar os parâmetros da rede a fim de encontrar a estrutura mais adequada ao reconhecimento dos padrões da base de dados. Já a segunda, objetiva ajustar os pesos da rede utilizando as técnicas de inteligência de enxames, em substituição ao *Rprop*. Em ambas as arquiteturas, a Otimização por Enxame de Partículas (PSO) foi aplicada, porém outras técnicas também poderiam ser utilizadas. Para a Arquitetura I, os resultados permitiram concluir que a inicialização das partículas é importante para tender as soluções resultantes para a estrutura da LIPNet, ou da *PyraNet*, versão especializada da primeira na qual o efeito inibitório é nulo. Além disso, chegou a identificar taxas melhores que a do trabalho original. Para a Arquitetura II, o PSO não foi uma boa solução, pois convergiu para mínimos locais, obtendo resultados piores que o método de treinamento original com *Rprop*.

Palavras-chave: LIPNet, PSO, *Particle Swarm Optimization*, Enxame, Receptivo, Inibitório, Campos, *Rprop*.

Abstract

The LIPNet is a pyramidal neural network with lateral inhibition used in computer vision to recognize patterns. This network is based on the concept of receptive and inhibitory fields from the human visual system and is formed by two layers: the first two dimensional in which the feature extraction is implied, and the second one-dimensional, responsible for classification of the input pattern. However, the training method of this network is the *Rprop*, which is likely to stagnate at local minima. To work around this problem, this monograph proposed techniques using swarm intelligence for optimization. Therefore, two architectures were defined: Architectures I and II. The first aims at optimizing the network parameters in order to find the most appropriate pattern recognition structure. The second, objectively adjust the weights of the network using swarm intelligence techniques, replacing the *Rprop*. In both architectures, the Particle Swarm Optimization (PSO) was applied, but other techniques could also be used. For Architecture I, the results showed that the initialization of the particles is important for the resulting solutions tend to structure LIPNet, or *PyraNet*, specialized version of the first in which the inhibitory effect is nil. Furthermore, it identify better than the original work rates. For Architecture II, the PSO was not a good solution because it converged to local minima, getting worse than the original training method with *Rprop* results.

Keywords: LIPNet, PSO, Particle Swarm Optimization, Swarm, Receptive, Inhibitory, Fields, Rprop.

Sumário

1	Introdução	1
1.1	Qualificação do Problema	1
1.2	Objetivos	2
1.2.1	Objetivos Específicos	2
1.3	Estrutura do Documento	2
2	Fundamentação teórica	3
2.1	Introdução	3
2.2	LIPNet	3
2.2.1	Sistema visual humano e inspiração biológica	3
2.2.1.1	Estrutura do olho humano	3
2.2.1.2	Campos receptivos e inibitórios	4
2.2.2	Trabalhos relacionados à LIPNet	5
2.2.3	Arquitetura da LIPNet	6
2.2.4	Treinamento da LIPNet	8
2.2.5	Considerações	10
2.3	Otimização por enxame de partículas	10
3	Treinamento da LIPNet com Enxame de Partículas	12
3.1	Introdução	12
3.2	Arquitetura I	12
3.3	Arquitetura II	14
4	Experimentos e Resultados	17
4.1	Introdução	17
4.2	Arranjo experimental	17
4.2.1	Arquitetura I	18
4.2.1.1	Definição dos experimentos para Arquitetura I	18
4.2.1.2	Execução dos experimentos para Arquitetura I	18
4.2.2	Arquitetura II	19
4.2.2.1	Definição dos experimentos para Arquitetura II	19
4.2.2.2	Execução dos experimentos para Arquitetura II	19
4.3	Resultados	19
4.3.1	Arquitetura I	20
4.3.1.1	Considerações sobre o tempo de simulação na Arquitetura I	21
4.3.2	Arquitetura II	22
5	Conclusões e Trabalhos Futuros	29
5.1	Conclusões	29
5.2	Trabalhos Futuros	29
	Referências	30

	APÊNDICE A – Soluções encontradas nas simulações da Arquitetura I	32
A.1	Experimento A da Arquitetura I	32
A.2	Experimento B da Arquitetura I	33
A.3	Experimento C da Arquitetura I	33

Lista de ilustrações

Figura 1 – Corte transversal do olho humano.	4
Figura 2 – Grade de Hermann.	5
Figura 3 – Arquitetura da LIPNet. (a) ilustra uma visão geral da rede, composta por camadas 2D na base e camadas 1D no topo. (b) mostra o relacionamento entre os neurônios dos campos receptivos e inibitórios da camada 2D. [Fonte: extraído de (FERNANDES, 2013)]	7
Figura 4 – Topologia Local do PSO.	11
Figura 5 – Representação da partícula na Arquitetura I.	12
Figura 6 – Relação entre entradas e saídas na Arquitetura I.	13
Figura 7 – Representação da partícula na Arquitetura II.	14
Figura 8 – Relação entre entradas e saídas na Arquitetura II.	16
Figura 9 – Exemplos de imagens do MIT CBCL Face Database: (a) faces de treinamento, (b) faces de teste, (c) não-faces de treinamento e (d) não-faces de teste.[Fonte: extraído de Fernandes (2013)]	17
Figura 10 – <i>Boxplot</i> da AUC para as melhores soluções encontradas nos Experimentos A, B e C da Arquitetura I. A linha tracejada indica a média da melhor solução encontrada por Fernandes (2013).	20
Figura 11 – <i>Boxplot</i> que ilustra a AUC obtida para 10 simulações da LIPNet com a configuração da melhor solução entre todos Experimentos da Arquitetura I (ARQ-I) e a melhor solução encontrada por Fernandes (2013)(Original).	21
Figura 12 – Comparação da curva de convergência da melhor partícula do PSO entre os Experimentos A, B e C da Arquitetura I.	22
Figura 13 – <i>Boxplot</i> do tempo médio de $q_{fit} = 10$ treinamentos em cada avaliação de <i>fitness</i> na Arquitetura I.	23
Figura 14 – <i>Boxplot</i> da AUC obtida para dois critérios de parada da otimização da Arquitetura II no Experimento A: Quantidade de Iterações (QI) e Validação Cruzada (VC).	24
Figura 15 – <i>Boxplot</i> da AUC obtida para dois critérios de parada da otimização da Arquitetura II no Experimento B: Quantidade de Iterações (QI) e Validação Cruzada (VC).	24
Figura 16 – Curva de convergência versus validação cruzada para o Experimento A da Arquitetura II.	25
Figura 17 – Curva de convergência versus validação cruzada para o Experimento B da Arquitetura II.	25
Figura 18 – <i>Boxplot</i> da AUC que compara o resultado obtido no Experimento A com a atualização dos pesos na proposta da Arquitetura II (ARQ-II) com o método original da LIPNet (<i>Rprop</i>).	26
Figura 19 – <i>Boxplot</i> da AUC que compara o resultado obtido no Experimento B com a atualização dos pesos na proposta da Arquitetura II (ARQ-II) com o método original da LIPNet (<i>Rprop</i>).	26
Figura 20 – Curva ROC do Experimento A da Arquitetura II.	27
Figura 21 – Curva ROC do Experimento B da Arquitetura II.	28
Figura 22 – Soluções encontradas para o Experimento A - Arquitetura I.	34
Figura 23 – Soluções encontradas para o Experimento B - Arquitetura I.	35
Figura 24 – Soluções encontradas para o Experimento C - Arquitetura I.	36

Lista de tabelas

Tabela 1 – Notações para descrição da rede LIPNet.	8
Tabela 2 – Precedência e limites da partícula na Arquitetura I.	14
Tabela 3 – Experimentos para inicialização das partículas na Arquitetura I.	18

Lista de Algoritmos

1	Pseudocódigo do algoritmo da fase <i>forward</i> da LIPNet.	9
2	Pseudocódigo do algoritmo de treinamento da LIPNet.	9
3	Pseudocódigo - PSO	11
4	Pseudocódigo - LIPNet-PSO-I.	15
5	Pseudocódigo - LIPNet-PSO-II.	16

Lista de abreviaturas e siglas

AUC	<i>Area Under the Curve</i>
CBCL	<i>Center for Biological and Computational Learning</i>
CE	<i>Cross Entropy</i>
CNN	<i>Convolutional Neural Network</i>
FEN	<i>Feature Extraction Network</i>
FEN+MLP	<i>Feature Extraction Network with Multilayer Perceptron</i>
FSS-II	<i>Fish School Search II</i>
LIPNet	<i>Lateral Inhibition Pyramidal Neural Network</i>
MLP	<i>Multilayer Perceptron</i>
PSO	<i>Particle Swarm Optimization</i>
PyraNet	<i>Pyramidal Neural Network</i>
QI	<i>Quantidade de Iterações</i>
RNA	Redes Neurais Artificiais
ROC	<i>Receiver Operating Characteristic</i>
Rprop	<i>Resilient Propagation</i>
VC	<i>Validação Cruzada</i>

1 Introdução

Neste trabalho de conclusão de curso, foram propostos modelos de treinamento da rede neural LIPNet (do inglês, *Lateral Inhibition Pyramidal Neural Network*) utilizando inteligência de enxames. Esta rede foi desenvolvida para a classificação de imagens, baseada na combinação da rede *PyraNet* (do inglês, *Pyramidal Neural Network*) com o conceito de campos inibitórios, presentes no sistema visual humano. Neste capítulo, é apresentada a introdução deste documento, que está organizada em 3 seções. Na Seção 1.1, é apresentada a motivação para a realização deste trabalho. Em seguida, na Seção 1.2, são apresentados os objetivos gerais e específicos e a hipótese de como o problema poderá ser solucionado. Por fim, na Seção 1.3, é descrita a estrutura do restante do documento.

1.1 Qualificação do Problema

É crescente a quantidade de aplicações que envolvem visão computacional: detecção de face (ROWLEY; BALUJA; KANADE, 1998), caracteres ópticos (GANIS; WILSON; BLUE, 1998), escrita (LECUN et al., 1998), entre outros. Esta área é promissora, pois incrementa facilidades no cotidiano das pessoas ao introduzir no sistema computacional características (ainda que limitadas) do sistema visual humano.

Paralelamente, as redes neurais artificiais encontraram aplicações em diversas áreas, dentre elas, clusterização de dados, compressão de dados, aproximação de funções e, inclusive, reconhecimento de padrões (FUKUSHIMA; MIYAKE; ITO, 1983). Arquitetura profunda é um conceito estreitamente ligado à aplicação das redes neurais e corresponde ao aprendizado de máquina em múltiplos níveis de abstração. No caso do reconhecimento de padrões, uma das vantagens das redes neurais de arquitetura profunda, por exemplo, é a possibilidade de elaborar métodos de reconhecimento independentes da extração de características do objeto estudado (FUKUSHIMA; MIYAKE; ITO, 1983; LECUN et al., 1989). Desta forma, o responsável por desenvolver o sistema de reconhecimento não precisa se preocupar com formas ou relacionamentos geométricos entre os elementos do padrão em estudo.

Phung e Bouzerdoum (2007) propuseram a PyraNet, inspirada no conceito de campos receptivos presente no sistema visual humano. Ela é estruturada em duas camadas: a primeira bidimensional, utilizada para efetuar a extração das características e reduzir a dimensionalidade dos dados; a segunda unidimensional, responsável pelo processo de reconhecimento.

Segundo Mao e Massaquoi (2007), também embasados na visão humana, a utilização de campos inibitórios pode tornar as redes neurais mais estáveis e eficientes. Com isso, Fernandes (2013) reformularam a PyraNet, incluindo o mecanismo de inibição lateral na mesma. De fato, a nova rede, nomeada LIPNet, apresentou bons resultados quando comparada a outras técnicas do estado-da-arte. Apesar das claras vantagens, as técnicas utilizadas para o treinamento da rede neural são baseadas em gradiente descendente e são susceptíveis a mínimos locais (RIEDMILLER; BRAUN, 1993; WANG et al., 2006).

Como ocorre com a maioria dos problemas reais, o espaço de busca pode apresentar uma grande quantidade de mínimos locais. Dessa maneira, a estratégia adotada na LIPNet do estado-da-arte pode ser melhorada ao utilizar inteligência de enxames para obter maior exploração do espaço de busca e, conseqüentemente, melhores taxas de reconhecimento. A inteligência de enxames é uma abordagem inspirada na sociedade de organismos simples que se comunicam e cooperam entre si, e já mostrou que pode ser usada para treinamento de outros tipos de redes neurais com sucesso (TEIXEIRA et al., 2008; SANTOS; VALENÇA; BASTOS-FILHO,

2012).

1.2 Objetivos

Este trabalho visa utilizar técnicas de inteligência de enxames a fim de aperfeiçoar os resultados obtidos com a LIPNet. A técnica escolhida para a análise é a Otimização por Enxame de Partículas (PSO - do inglês, *Particle Swarm Optimization*) (KENNEDY; EBERHART, 1995; BRATTON; KENNEDY, 2007), por ser uma técnica adequada para otimização de variáveis contínuas em espaços hiperdimensionais. Como resultado obtido, é proposta uma ferramenta computacional para o reconhecimento de padrões e fundamentada numa abordagem híbrida da LIPNet com conceitos de inteligência de enxames.

Portanto, este projeto tem por objetivo geral desenvolver uma técnica híbrida voltada para o reconhecimento de padrões. A tecnologia envolvida no reconhecimento é a versão hibridizada da rede neural LIPNet por meio da aplicação de algoritmos de inteligência de enxames.

1.2.1 Objetivos Específicos

Por objetivos específicos, destacam-se:

- Propor diferentes arquiteturas para a integração entre a LIPNet e o algoritmo de inteligência de enxames, tais como, otimizar os parâmetros da rede ou a forma com a qual os seus pesos são atualizados;
- Analisar o desempenho do algoritmo PSO para o treinamento da LIPNet;
- Verificar as taxas de acerto e erro encontradas para a detecção de faces no banco de imagens de faces (HEISELE; POGGIO; PONTIL, 2000).

1.3 Estrutura do Documento

Esta monografia está organizada em 5 capítulos. O Capítulo 2 apresenta a fundamentação teórica, destacando a LIPNet, rede neural base usada neste trabalho, desde a sua inspiração biológica até o modelo matemático, e a otimização por enxame de partículas. Em seguida, o Capítulo 3 apresenta os métodos propostos. No Capítulo 4, os experimentos são detalhados e os seus respectivos resultados são apresentados. Por fim, o Capítulo 5 resume as principais contribuições e sugere alguns trabalhos futuros.

2 Fundamentação teórica

2.1 Introdução

Este capítulo objetiva detalhar os conceitos básicos envolvidos no desenvolvimento deste trabalho. Na Seção 2.2, a LIPNet é detalhada. A Seção 2.3 explana resumidamente o algoritmo de Otimização por Enxame de Partículas básico.

2.2 LIPNet

A visão computacional é um ramo da ciência que visa introduzir nas máquinas a capacidade de interpretar imagens analogamente ao sistema visual humano. As aplicações incluem detecção de face (ROWLEY; BALUJA; KANADE, 1998), caracteres ópticos (GANIS; WILSON; BLUE, 1998), escrita (LECUN et al., 1998), entre outros. Nesse ramo, as redes neurais artificiais constituem técnicas que são amplamente utilizadas. Dentre elas, está a LIPNet, recentemente desenvolvida por Fernandes (2013). Esta rede introduz características do sistema visual humano no sistema computacional e embasa este trabalho. Portanto, este capítulo visa descrevê-la.

Na Subseção 2.2.1, é apresentada a inspiração biológica. Na Subseção 2.2.2, alguns estudos que antecederam a LIPNet são apresentados resumidamente. Na Subseção 2.2.3, a arquitetura da rede é detalhada. O método de treinamento é sumarizado na Subseção 2.2.4. E, por fim, a Subseção 2.2.5 apresenta algumas considerações a respeito da LIPNet.

2.2.1 Sistema visual humano e inspiração biológica

Assimilar informações do ambiente natural é primordial para os seres vivos, capacitando-os a fazer uso dos recursos externos como forma de sobrevivência. Nesse contexto, o sistema visual humano atua como interface com o mundo exterior ao permitir que a luz visível captada seja interpretada e esta informação, aproveitada de alguma maneira em benefício próprio.

É interessante notar a introdução de aspectos inerentes à biologia do sistema visual propriamente dito nos modelos de visão computacional. Isso ocorre porque a visão computacional é uma área de pesquisa que visa introduzir nas máquinas a habilidade humana de interpretar imagens reconhecendo ou identificando padrões. Dessa forma, se a intenção é imitar o sistema visual humano, é natural, por vezes, considerar suas características nos modelos computacionais.

Neste cenário de inspirações biológicas, surgem importantes conceitos como o de campos receptivos e inibitórios presentes no sistema visual humano e que compõem a base da LIPNet. Portanto, uma breve explicação sobre esse sistema é feita nas subseções a seguir, a fim de aperfeiçoar o conhecimento sobre os conceitos biológicos básicos envolvidos na rede neural em questão.

2.2.1.1 Estrutura do olho humano

Gonzalez e Woods (2010) descrevem a anatomia e fisiologia do olho humano, ilustrado na Figura 1.

A córnea é uma membrana transparente e resistente que reveste a parte anterior do olho e refrata a luz. Após passar pela córnea, o raio luminoso atravessa a câmara anterior até passar pela pupila, que é

uma pequena abertura localizada na íris, estrutura circular pigmentada cuja contração e expansão controla a quantidade de luz que entra no olho. O corpo ciliar é responsável pela acomodação do cristalino, o qual é uma lente convergente e gelatinosa que focaliza o raio luminoso na retina. Nesta, a presença de receptores discretos de luz transforma a informação luminosa refletida em sinais elétricos.

Os receptores de luz localizados na retina são cones e bastonetes. Os primeiros se concentram na fóvea, porção central da retina. Eles permitem a distinção entre as diferentes cores. Já os bastonetes, são sensíveis a baixos níveis de iluminação, não identificam cores e estão envolvidos na percepção do campo de visão geral da imagem. A luz que incide sobre os cones e bastonetes provoca uma reação química produzindo

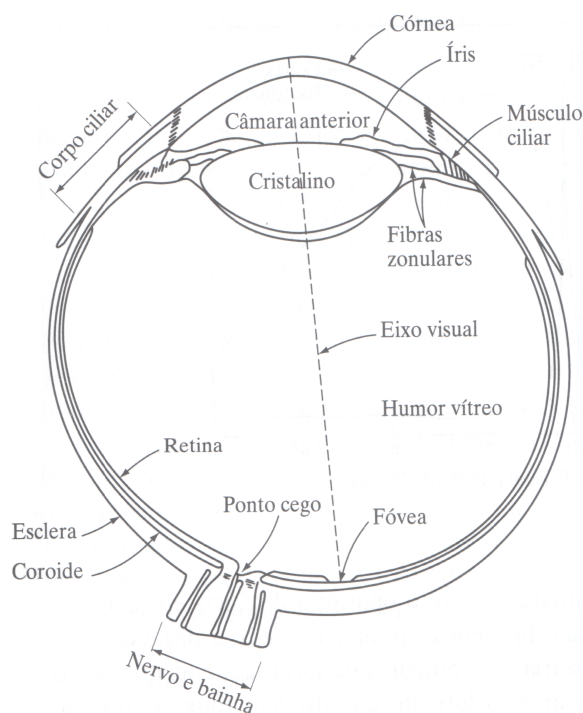


Figura 1 – Corte transversal do olho humano.

[Fonte: extraído de (GONZALEZ; WOODS, 2010)]

impulsos elétricos que são transmitidos às células bipolares. Estas, por sua vez, se ligam ao axônio das células ganglionares, as quais formam o nervo óptico, responsável por levar os impulsos elétricos até o cérebro. Dessa forma, cada célula ganglionar recebe os sinais de regiões específicas da retina, denominadas campos receptivos, que, por sua vez, abrange milhares de fotorreceptores.

2.2.1.2 Campos receptivos e inibitórios

Hartline (1938) definiu campo receptivo como uma região restrita do campo visual na qual o estímulo luminoso pode levar a respostas elétricas das células ganglionares. Após a definição de Hartline (1938), este conceito foi estendido para outras células sensitivas, pois um comportamento similar foi identificado nos sistemas auditório, somatossensório e visual (HUBEL, 1963). Levine e Shefner (2000) generalizaram este conceito definindo o campo receptivo como uma área na qual a presença de estímulo apropriado deve conduzir a resposta de um neurônio sensitivo.

Rizzolati e Camarda (1975) identificaram que a aplicação de outro estímulo simultâneo ao do campo receptivo também pode afetar a saída do neurônio. Neste caso, uma região atua com um efeito inibitório sobre o neurônio, daí o conceito de campos inibitórios. Um exemplo clássico da influência do campo inibitório é ilustrado na Figura 2. A Grade de Hermann (1870) é formada por linhas brancas verticais e horizontais

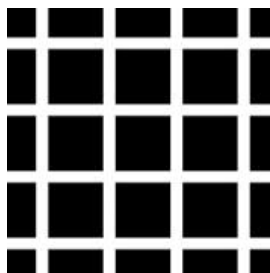


Figura 2 – Grade de Hermann.

[Fonte: extraído de (HERMANN, 1870)]

sobrepostas e equidistantes entre si. Ao olhar para a grade, manchas em tons de cinza podem ser observadas nas interseções. Isso ocorre porque os neurônios são ativados ao detectar alta luminosidade no seu campo receptivo e, simultaneamente, exercem um efeito inibitório sobre os neurônios inclusos no seu raio de inibição. Uma vez que a interseção é cercada por mais pontos com maior intensidade do que o ponto central entre interseções, o somatório do efeito inibitório nos pontos de interseção é maior do que o impulso de ativação do próprio neurônio. Logo, a inibição causa um efeito negativo sobre a luminosidade da interseção, evidenciando as manchas.

2.2.2 Trabalhos relacionados à LIPNet

As Redes Neurais Artificiais (RNAs) constituem uma técnica de inteligência computacional que utiliza elementos básicos de computação não linear (neurônios) organizados em redes, de maneira análoga àquela que os neurônios estão interconectados no cérebro humano (GONZALEZ; WOODS, 2010). As redes neurais artificiais encontraram aplicações em diversas áreas, dentre elas, clusterização de dados, classificação de padrões, previsão, compressão de dados e aproximação de funções.

Paralelamente, é crescente a aplicação de redes neurais no ramo da visão computacional como em: detecção de face (ROWLEY; BALUJA; KANADE, 1998), identificação de caracteres ópticos (GANIS; WILSON; BLUE, 1998), reconhecimento de dígitos manuscritos (LECUN et al., 1998), entre outros. Esta área é promissora, pois incrementa facilidades no cotidiano das pessoas ao introduzir no sistema computacional, características (ainda que limitadas) do sistema visual humano.

É natural, no início do aprendizado, considerar exemplos simples para posteriormente serem generalizados para situações mais complexas. Ou seja, o aprendizado é estreitamente ligado à abstração de conceitos, o que sugere a existência de uma hierarquia de responsabilidades no cérebro (HUBEL, 1963). Esse modelo de aprendizado de máquina em múltiplos níveis é chamado de aprendizagem profunda e tipicamente envolve redes neurais. Uma das vantagens das RNAs de arquitetura profunda, por exemplo, é a possibilidade de elaborar métodos de reconhecimento independentes da extração de características do objeto estudado. Desta forma, o responsável por desenvolver o sistema de reconhecimento não precisa se preocupar com formas ou relacionamentos geométricos entre os elementos do padrão em estudo.

A LIPNet é uma rede neural piramidal com inibição lateral, a qual foi inspirada em modelos de arqui-

tetura profunda, como o Neocognitron (FUKUSHIMA; MIYAKE; ITO, 1983), Rede Neural Convolutacional (*Convolutional Neural Network*, CNN) (LECUN et al., 1989), Rede de Extração de Características com MLP (*Feature Extraction Network with Multilayer Perceptron*, FEN+MLP) (PEREZ et al., 2003) e *PyraNet* (*Pyramidal Neural Network*) (PHUNG; BOUZERDOUM, 2007).

O Neocognitron inspirou-se no conceito de campos receptivos e inibitórios do sistema visual humano e os seus múltiplos níveis são alternadamente compostos por células simples e complexas. O reconhecimento é iniciado com a extração de características locais dos padrões de entrada na primeira camada e, gradualmente, as características globais são identificadas conforme a passagem de nível na RNA. A CNN incorpora as características do Neocognitron, acrescentando-se ao topo uma rede completamente conectada, responsável pela classificação dos padrões. Dessa forma, as características extraídas aos longo dos níveis da rede são combinadas para a extração de características mais complexas. Perez et al. (2003) propuseram um modelo com algoritmo genético para definir a melhor configuração dos campos receptivos e inibitórios na FEN+MLP. A FEN, assim como as demais redes apresentadas nesta seção, baseia-se no Neocognitron. Por fim, a *PyraNet* é estruturada em duas partes: a primeira bidimensional e em formato piramidal, utilizada para efetuar a extração das características e reduzir a dimensionalidade dos dados; a segunda unidimensional, responsável pelo processo de reconhecimento.

Estudos apontam que a utilização de campos inibitórios pode tornar as redes neurais mais estáveis e eficientes (COULTRIP; GRANGER; LYNCH, 1992; MAO; MASSAQUOI, 2007). Com isso, Fernandes, Cavalcanti e Ren (2013) reformularam a *PyraNet*, acrescentando-lhe o mecanismo de inibição lateral. De fato, a rede proposta, nomeada LIPNet, apresentou bons resultados quando comparada a outras técnicas do estado-da-arte.

2.2.3 Arquitetura da LIPNet

Do mesmo modo que a *PyraNet*, a LIPNet é estruturada em duas camadas, conforme é apresentado na Figura 3. Na Figura 3.a, observa-se a estrutura 2D (2 dimensões) da primeira camada. Nesta, cada neurônio está associado a uma região, denominada campo receptivo, com neurônios da camada anterior, de modo que uma pirâmide é formada. Essa camada possibilita a extração implícita das características do padrão apresentado e a redução da dimensionalidade. No topo da estrutura está a camada 1D (1 dimensão), cuja finalidade é classificar as imagens a partir das características obtidas na camada inferior. A interface entre as duas camadas se dá apenas pela reorganização dos neurônios de uma matriz para um vetor.

Na Figura 3.b, destaca-se o relacionamento entre os neurônios da camada 2D. Um neurônio na posição (u, v) da camada l liga-se aos neurônios pertencentes ao seu campo receptivo $R_{u,v}^l$, na camada $l - 1$. A sobreposição entre campos receptivos adjacentes é possível e é denotada pelo *overlap* o . A inibição lateral é representada na Figura 3.b pelas setas, nas quais o neurônio da origem interfere negativamente na saída do destino. Este tipo de efeito ocorre apenas entre neurônios da mesma camada que estão inclusos no campo inibitório de tamanho h . A força ψ com que os vizinhos influenciam a saída de um neurônio é dependente do peso inibitório σ e é calculada usando a Equação (2.1)

$$\psi^l = \frac{\sigma^l}{(2h^l + 1)^2 - 1}. \quad (2.1)$$

Cada neurônio 2D está associado diretamente a um peso w e uma polarização externa b . A saída y do neurônio é dada por

$$y_{u,v}^l = f(\chi_{u,v}^l - \iota_{u,v}^l), \quad (2.2)$$

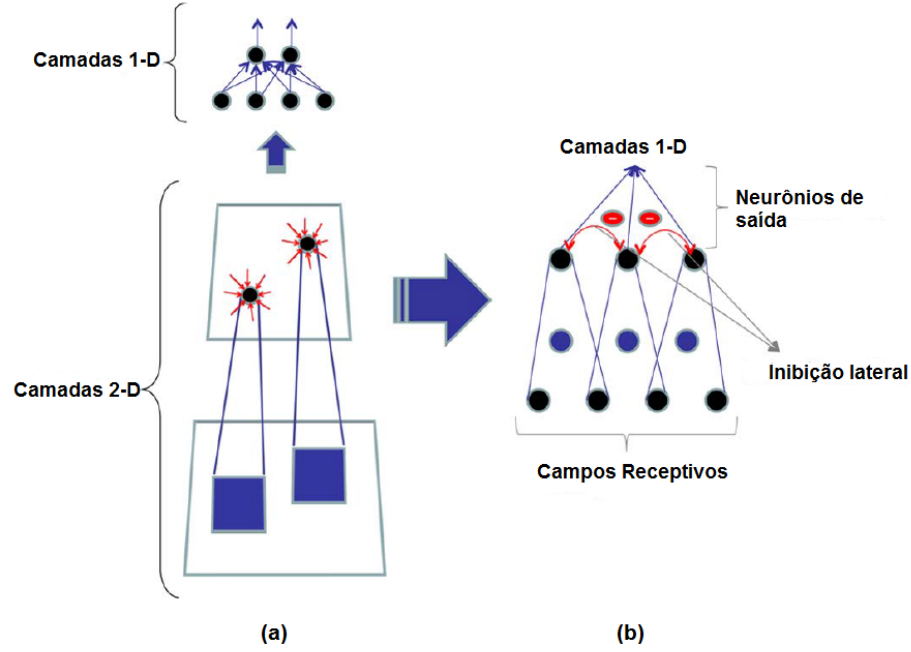


Figura 3 – Arquitetura da LIPNet. (a) ilustra uma visão geral da rede, composta por camadas 2D na base e camadas 1D no topo. (b) mostra o relacionamento entre os neurônios dos campos receptivos e inibitórios da camada 2D. [Fonte: extraído de (FERNANDES, 2013)]

onde $f(\cdot)$ é a função de ativação, $\chi_{u,v}^l$ e $l_{u,v}^l$ indicam, respectivamente, o estímulo excitatório, Equação (2.3), e inibitório, Equação (2.4), do neurônio (u, v) da camada l . O conjunto de neurônios pertencentes ao campo receptivo do neurônio (u, v) da camada l é indicado por $R_{u,v}^l$.

$$\chi_{u,v}^l = \sum_{i,j \in R_{u,v}^l} w_{i,j}^l y_{i,j}^{l-1} + b_{u,v}^l, \quad (2.3)$$

$$l_{u,v}^l = \psi^l \left[\left(\sum_{i=u-h}^{u+h} \sum_{j=v-h}^{v+h} f(\chi_{i,j}^l) \right) - f(\chi_{u,v}^l) \right]. \quad (2.4)$$

Já para a camada 1D, a saída dos neurônios é definida segundo a seguinte equação:

$$y_n^l = f(s_n^l), \quad (2.5)$$

onde s_n^l é o somatório da entrada do neurônio n e é expressa por:

$$s_n^l = \sum_{m=1}^{N_{l-1}} w_{m,n}^l y_m^{l-1} + b_n^l. \quad (2.6)$$

A entrada da rede é uma imagem k , de modo que cada pixel seja a saída de um neurônio da primeira camada 2D. Já a saída da rede deve possuir uma quantidade de neurônios igual ao número de classes do problema. Como esta rede neural é de aprendizado supervisionado, então o objetivo é reduzir o erro da saída calculada em relação à desejada. Para tanto, Fernandes, Cavalcanti e Ren (2013) utilizaram o método da

entropia cruzada (CE - do inglês, *Cross-Entropy*) (BISHOP, 2007) como função de otimização para ajuste dos pesos da rede neural. Portanto, a probabilidade p da imagem k pertencer à classe indicada pelo neurônio de saída n é dada por

$$p_n^k = \frac{\exp(y_n^{L_{1D},k})}{\sum_{i=1}^{N_{L_{1D}}} \exp(y_i^{L_{1D},k})}, \quad (2.7)$$

onde $N_{L_{1D}}$ é a quantidade de neurônios da última camada L_{1D} e a saída do neurônio n da última camada L_{1D} para a imagem k é representada por $y_n^{L_{1D},k}$.

As notações utilizadas são resumidas na Tabela 1 e o Algoritmo 1 sintetiza a etapa *forward*, que antecede o ajuste dos pesos da LIPNet.

Tabela 1 – Notações para descrição da rede LIPNet.

[Fonte: adaptado de (FERNANDES, 2013)]

Símbolo	Descrição
L_{2D}	Última camada piramidal 2D
L_{1D}	Última camada 1D
l	Camada da rede neural, onde $l \in \{1, \dots, L_{2D}\}$ para as camadas 2D e $l \in \{L_{2D} + 1, \dots, L_{1D}\}$ para as camadas 1D
H^l	Altura da camada 2D l
W^l	Largura da camada 2D l
N^l	Tamanho da camada 1D l
r^l	Tamanho dos campos receptivos da camada 2D l
o^l	Tamanho da sobreposição entre os campos receptivos adjacentes da camada 2D l
h^l	Tamanho do campo inibitório da camada 2D l
σ^l	Peso inibitório da camada 2D l
ψ_l	Força da inibição lateral na camada l
$\chi_{u,v}^l$	Estímulo excitatório do neurônio da posição (u, v) da camada 2D l
$w_{i,j}^l$	Peso associado do neurônio da posição (i, j) da camada $l - 1$ para a camada 2D l
$y_{u,v}^l$	Saída do neurônio da posição (u, v) da camada 2D l
$b_{u,v}^l$	<i>Bias</i> do neurônio (u, v) da camada 2D l
$\iota_{u,v}^l$	Estímulo inibitório do neurônio da posição (u, v) da camada 2D l
f	Função de ativação
y_m^l	Saída do neurônio da posição m na camada 1D l
s_n^l	Somatório de entrada do neurônio n da camada 1D l
$w_{m,n}^l$	Peso sináptico entre o neurônio da posição m da camada 1D $l - 1$ e o neurônio da posição n da camada 1D l
$R_{u,v}^l$	Conjunto de neurônios que pertencem ao campo receptivo do neurônio da posição (u, v) da camada 2D l
p_n^k	Probabilidade <i>a posteriori</i> para a classe associada ao neurônio n na camada de saída de rede para a imagem de entrada k
d_n^k	Saída desejada do neurônio n para a imagem de entrada k

2.2.4 Treinamento da LIPNet

A fase *backward* da LIPNet inicia-se com o cálculo da sensibilidade para a camada 1D. Tal processo se dá de modo análogo ao da tradicional MLP. Posteriormente, a sensibilidade dos neurônios da camada 2D ocorre em três etapas para cada camada l :

1. cálculo da sensibilidade do neurônio com relação à camada $l + 1$. Esta etapa considera a influência dos campos receptivos;

Algoritmo 1: Pseudocódigo do algoritmo da fase *forward* da LIPNet.

Entrada: Imagem k ;
Saída: Probabilidade *a posteriori* p_n^k da imagem k pertencer a cada classe conhecida n ;

- 1 Inicialize a primeira camada 2D ($l = 0$) de modo que cada pixel da imagem k corresponda à saída de um neurônio dessa camada;
- 2 **para cada** camada 2D, $l = 1 \rightarrow L_{2D}$ **faça**
- 3 **para cada** neurônio (u, v) , $u \rightarrow \{1..H^l\}$ e $v \rightarrow \{1..W^l\}$ **faça**
- 4 | $y_{u,v}^l \leftarrow Eq.(2.2)$;
- 5 Rearrume a última camada 2D ($l = L_{2D}$) em um vetor unidimensional, representando a primeira camada 1D;
- 6 **para cada** camada 1D, $l = 1 \rightarrow L_{1D}$ **faça**
- 7 **para cada** neurônio m , $m \rightarrow \{1..N^l\}$ **faça**
- 8 | $y_m^l \leftarrow Eq.(2.5)$;
- 9 **para cada** neurônio m da última camada 1D ($l = L_{1D}$) **faça**
- 10 | $p_n^k \leftarrow Eq.(2.7)$;

2. cálculo da sensibilidade do neurônio com relação à própria camada l . Esta etapa é uma adaptação para considerar a influência dos campos inibitórios no ajuste dos pesos;
3. cálculo da sensibilidade do neurônio. Definida com base na subtração do resultado obtido na primeira etapa pelo da segunda.

O gradiente dos pesos e polarizações externas são definidos para, por fim, os pesos serem ajustados utilizando o método de treino *Resilient Propagation (Rprop)* (RIEDMILLER; BRAUN, 1993). Esse método é de retropropagação do erro, diferindo do Gradiente Descendente pelo fato do *Rprop* fazer sua atualização com base somente no sinal do erro. Por isso, ele é um método conhecido pela rápida convergência e consequente baixo custo computacional. Vale ressaltar que o modelo matemático tradicional do *backpropagation* não é ideal para aplicação na LIPNet, pois o primeiro é específico para um arranjo unidimensional de neurônios, enquanto que a LIPNet possui uma estrutura bidimensional. Além disso, Fernandes (2013) também analisou o método do *backpropagation* adaptado à LIPNet e obteve piores resultados que o *Rprop*. Apesar das vantagens, o *Rprop* é um método sujeito à estagnação em mínimos locais (WANG et al., 2006), o que compromete os resultados da LIPNet. A solução proposta no Capítulo 3 pretende avaliar se este problema ocorre e se é possível resolvê-lo.

O Algoritmo 2 resume o treinamento da LIPNet.

Algoritmo 2: Pseudocódigo do algoritmo de treinamento da LIPNet.

Entrada: Imagens e suas respectivas classes;
Saída: LIPNet treinada;

- 1 **enquanto** critério de parada não é atingido **faça**
- 2 **para cada** Imagem k e sua respectiva saída desejada d_n^k **faça**
- 3 | Calcule a saída da rede para a imagem k , segundo a etapa *forward* (Algoritmo 1);
- 4 | Acumule o gradiente erro de cada neurônio n , seguindo a etapa *backward* (FERNANDES, 2013);
- 5 **para cada** peso w e bias da LIPNet **faça**
- 6 | Atualize peso utilizando o método *Rprop* (RIEDMILLER; BRAUN, 1993);

2.2.5 Considerações

Os experimentos realizados com a LIPNet não analisaram a profundidade da rede. Os estudos feitos por Fernandes (2013) consideram usualmente duas camadas piramidais, pois as imagens do banco utilizado são pequenas e, conseqüentemente, a adição de novas camadas não se mostraria eficaz.

Vale ressaltar que a *PyraNet* pode ser considerada uma especialização da LIPNet, na qual o efeito da inibição lateral é inexistente (tamanho do campo inibitório ou peso inibitório nulos). Este conceito é importante para entender do desenvolvimento deste trabalho.

2.3 Otimização por enxame de partículas

O PSO é uma técnica de Inteligência de Enxames, a qual se inspira na habilidade do comportamento em grupo de alguns animais. A partir da característica social que possibilita a comunicação entre os indivíduos, emerge um comportamento que conduz os agentes reativos simples para regiões ótimas do espaço de busca. Essa característica potencializa o encontro de soluções satisfatórias em problemas não-lineares e, especialmente no algoritmo original do PSO, com variáveis contínuas. O PSO foi proposto por Kennedy e Eberhart (1995) e é um dos mais referenciados algoritmos de inteligência de enxames. Baseado no comportamento do voo de pássaros, o PSO possui um mecanismo estocástico simples para guiar partículas em busca de uma solução ótima global. Para encontrar uma solução, é necessário definir uma função de avaliação (*fitness*) específica para o problema. Essa função representa o espaço de busca e é utilizada para guiar as partículas durante a otimização.

No PSO, cada partícula i representa uma possível solução para o problema em questão e é composta por três vetores de D dimensões: a posição da partícula no espaço de busca \vec{x}_i^t , a melhor posição que a partícula achou individualmente \vec{p}_i^t e a sua velocidade \vec{v}_i^t no instante de tempo t . O ajuste da posição \vec{x}_i^t é feito de acordo com a própria experiência partícula e com a do enxame como um todo. Primeiramente, atualiza-se a velocidade segundo a equação:

$$\vec{v}_i^{t+1} = \chi [\vec{v}_i^t + c_1 r_1 (\vec{p}_i^t - \vec{x}_i^t) + c_2 r_2 (\vec{n}_i^t - \vec{x}_i^t)], \quad (2.8)$$

onde \vec{n}_i^t representa a melhor posição encontrada na vizinhança da partícula i , χ é uma constante chamada fator de restrição, r_1 e r_2 são variáveis aleatórias geradas por uma função de distribuição de probabilidade uniforme no intervalo $[0, 1]$, c_1 e c_2 são, respectivamente, coeficientes de aceleração cognitivo e social. Para este trabalho, admitiu-se $\chi = 0,72984$ e $c_1 = c_2 = 2,05$, seguindo o estudo de Bratton e Kennedy (2007). Após a atualização da velocidade, a nova posição da partícula é dada por

$$\vec{x}_i^{t+1} = \vec{x}_i^t + \vec{v}_i^{t+1}, \quad (2.9)$$

para a variação de tempo igual a uma unidade.

A vizinhança da partícula é determinada conforme a topologia utilizada, definindo como se dá a comunicação no enxame. Uma das topologias básicas do PSO é a local, na qual cada partícula interage diretamente apenas com um número limitado de vizinhas. A Figura 4 ilustra essa topologia: os círculos representam os indivíduos que estão ligados aos vizinhos. Essa estrutura permite que a informação seja disseminada por todo enxame de forma lenta, favorecendo uma melhor exploração em amplitude do espaço de busca.

O Algoritmo 3 descreve o pseudocódigo básico do PSO, onde \vec{n}_{best} é a melhor posição encontrada por todo o enxame.

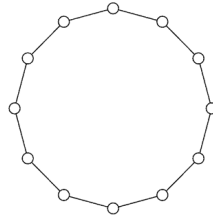


Figura 4 – Topologia Local do PSO.

Algoritmo 3: Pseudocódigo - PSO

```
1 Inicializa população com  $n$  partículas;  
2 enquanto condição de parada não for alcançada faça  
3   para cada partícula  $i$  faça  
4     Atualiza  $\vec{v}_i$ , usando Eq.(2.8);  
5     Atualiza  $\vec{x}_i$ , usando Eq.(2.9);  
6     Avalia fitness da partícula;  
7     Atualiza  $\vec{p}_i$ ;  
8   para cada partícula  $i$  faça  
9     Atualiza  $\vec{n}_i$ ;  
10    Atualiza  $\vec{n}_{best}$ ;  
11 Retorna  $n_{best}$ .
```

3 Treinamento da LIPNet com Enxame de Partículas

3.1 Introdução

Este capítulo apresenta o modelo proposto como alternativa ao método de treinamento da LIPNet: a Arquitetura I propõe uma otimização dos parâmetros da rede, de modo que o melhor resultado encontrado pode ser utilizado como entrada para a Arquitetura II. Esta visa definir os pesos da rede com uma técnica de inteligência de enxames, sem necessitar, portanto, do método *Rprop* para o treinamento. Neste trabalho ambas as arquiteturas utilizam o algoritmo PSO de topologia local definido na Seção 2.3 para otimizar a rede, porém, outros modelos também podem ser aplicados.

A Seção 3.2 detalha o primeiro modelo proposto por este estudo: Arquitetura I. A Seção 3.3 descreve o segundo modelo: Arquitetura II.

3.2 Arquitetura I

A Arquitetura I é proposta para identificar parâmetros da LIPNet que favoreçam o reconhecimento de determinado padrão.

A Figura 5 ilustra a estrutura da partícula. Neste caso, a otimização busca para cada camada 2D, a largura do campo receptivo r , *overlap* o , raio do campo inibitório h e peso inibitório σ . Para cada camada 1D, procura-se identificar o seu tamanho N . Além disso, as funções de ativação de todos os neurônios da camada 2D, f_{2D} , e da camada 1D, f_{1D} , também são otimizadas. Desse modo, a quantidade de camadas L_{2D} e L_{1D} são parâmetros do sistema. Já o tamanho H da camada 2D, l , é calculado com base nos tamanhos do campo receptivo r e *overlap* o conforme

$$H^l = \left\lceil \frac{H^{l-1} - o^l}{r^l - o^l} \right\rceil. \quad (3.1)$$



Figura 5 – Representação da partícula na Arquitetura I.

Vale ressaltar que, para essa rede, considerou-se que os campos receptivos e inibitórios são quadrados.

A Figura 6 resume a Arquitetura I, ilustrando o relacionamento entre as entradas e saídas. Primeiramente, o algoritmo de otimização recebe a quantidade de camadas L_{2D} e L_{1D} como entrada. Durante a avaliação do *fitness* das partículas, a LIPNet recebe os parâmetros definidos. Na LIPNet, ocorre o procedimento básico de inicialização, treino e teste, para, ao final, retornar a área sob a curva ROC (AUC^1 - do inglês, *Area Under the Curve*) (BRADLEY, 1997). Como a inicialização dos pesos no treinamento da LIPNet é aleatória, então duas ou mais avaliações com os mesmos parâmetros resultam em AUCs distintas. Para contornar esse problema, reduzindo a variabilidade dos resultados, cada avaliação de *fitness* é associada à média de q_{fit}

¹ Curva ROC (do inglês, *Receiver Operating Characteristic*) taxa de falso positivos de um classificador. Sua precisão é medida através da AUC, a qual pertence ao intervalo $[0, 1]$: quanto mais próximo de 1, melhor o resultado.

simulações da LIPNet.

Cada elemento da partícula pertence a diferentes espaços de busca: o campo receptivo, por exemplo,

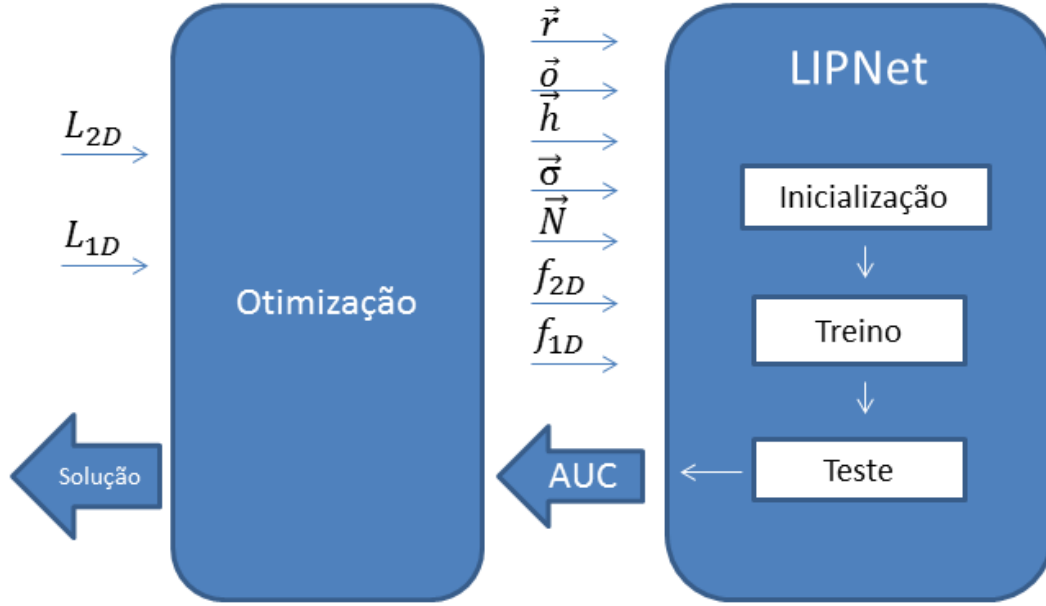


Figura 6 – Relação entre entradas e saídas na Arquitetura I.

pode assumir valores que não pertencem ao campo de busca das funções de ativação. Consequentemente, faz-se necessário o tratamento individualizado para cada parte da partícula. Além disso, há relações de dependência: o tamanho do *overlap* deve ser obrigatoriamente menor que o campo receptivo, por exemplo. Dessa forma, a atualização das velocidades e posições deve obedecer à ordem de precedência.

Para que a partícula seja válida, umas das verificações feitas é que o tamanho da camada 2D, H^l , seja maior ou igual ao tamanho mínimo H_{min}^l . Esta validação pretende atender à premissa de que a quantidade de camadas 2D, L_{2D} , seja respeitada para quaisquer partículas. Já que H^l é calculado a partir de r e o (ver Equação (3.1)). Então, obrigatoriamente, existe ao menos uma combinação de campo receptivo r e *overlap* o que viabilize a premissa e o tamanho mínimo do campo receptivo $r_{min} = 2$ simultaneamente. Caso contrário, se $r > r_{max}$ ou $o < o_{min}$, é impossível atingir L_{2D} camadas. Portanto, H_{min}^l é dado por

$$H_{min}^l = (r_{min})^{L_{2D}-l}. \quad (3.2)$$

Como o PSO é um algoritmo específico para espaços de busca contínuos, todos os elementos da partícula, excetuando o peso inibitório, são truncados. Ou seja, na otimização, a partícula pertence ao espaço contínuo, porém, ao avaliar o *fitness*, cada variável discreta v é arredondada para $\lfloor v \rfloor$.

A Tabela 2 destaca a precedência e os limites máximo e mínimo assumidos por cada elemento da partícula. O intervalo assumido é [Mínimo,Máximo), a fim de considerar o truncamento das variáveis discretas. O limite máximo de f_{2D} e f_{1D} é igual a dois, pois foram duas as funções de ativação consideradas: sigmóide logística e tangente hiperbólica.

Caso não fossem tratados como constantes no processo de otimização, L_{2D} e L_{1D} causariam perda da informação ao reduzir ou aumentar a quantidade de camadas. Isso porque partículas até então válidas, poderiam se tornar inválidas. Por exemplo, considere uma partícula que define uma estrutura de rede cuja

Tabela 2 – Precedência e limites da partícula na Arquitetura I.

Elemento	Precedência	Mínimo	Máximo
r^l	-	r_{min}	$r_{max} = H^{l-1} - H_{min}^l + 2$
o^l	r^l	$o_{min} = \left\lceil \frac{H_{min}^l r^l - H^{l-1}}{H_{min}^l - 1} \right\rceil$	$o_{max} = r^l$
h^l	o^l	$h_{min} = 0$	$h_{max} = (H^l - 1)/2 + 1$
$\sigma^l *$	-	$\sigma_{min} = 0.1$	$\sigma_{max} = 30$
$N^l *$	-	1	10
f_{2D} e f_{1D}	-	0	2

* O intervalo assumido é entre [Mínimo,Máximo]

última camada 2D tem tamanho igual a 1. Caso o número de camadas aumentasse, não seria possível definir outra solução baseada na anterior tal que $r_{min} = 2$. Logo, parte da partícula seria reinicializada perdendo parte, ou todo, conhecimento conquistado até então.

A Arquitetura I com PSO é sumarizada no Algoritmo 4, onde $v_i^{r,l}$, $v_i^{o,l}$, $v_i^{h,l}$, $v_i^{\sigma,l}$ e $v_i^{N,l}$ correspondem, respectivamente, à velocidade do tamanho do campo receptivo, *overlap*, campo inibitório, peso inibitório e tamanho da camada 1D para a partícula i e camada l .

3.3 Arquitetura II

A Arquitetura II visa substituir o método de ajuste dos pesos *Rprop* pelos pesos encontrados na otimização. Desse modo, o treinamento da rede é a própria otimização. Portanto, a partícula desta arquitetura é composta apenas pelos pesos e polarizações externas necessários à rede, como ilustra a Figura 7. Na figura, $w_{i,j}^l$ representa todos os pesos da camada 2D e $b_{i,j}^l$, todos as polarizações externas, onde $i \in \{1, \dots, H^l\}$, $j \in \{1, \dots, W^l\}$ e $l \in \{1, \dots, L_{2D}\}$. Já $w_{m,n}^l$ e b_m^l referem-se, respectivamente, aos pesos e polarizações externas da camada 1D, onde $m \in \{0, \dots, L_{1D} - 1\}$, $n \in \{1, \dots, L_{1D}\}$ e $l \in \{1, \dots, L_{1D}\}$.

**Figura 7 – Representação da partícula na Arquitetura II.**

A Figura 8 resume as entradas e saídas para a Arquitetura II. Neste caso, uma configuração da LIPNet é definida como entrada para a otimização. O objetivo da estrutura da LIPNet é caracterizar o tamanho da partícula e calcular a saída da rede para os pesos e polarizações externas otimizados. Portanto, após receber a solução da partícula como entrada, os pesos da LIPNet são atualizados para posterior avaliação da rede segundo um banco de imagens de treino. O resultado desta avaliação é o *fitness* da partícula em questão. Note que, diferentemente da Arquitetura I, a avaliação de *fitness* é determinística. Do mesmo modo que qualquer outro processo de otimização, a saída dessa arquitetura é a melhor solução encontrada.

Outra diferença em relação à Arquitetura I é que ambos, peso e polarização externa pertencem ao espaço de busca contínuo. Isto favorece o uso do PSO nesta arquitetura, já que não será necessário trabalhar com truncamentos, como ocorre no outro caso. A partícula foi limitada no intervalo $[10^{-8}, 100]$, pois verificou-se que este intervalo atende à maioria dos pesos utilizando a LIPNet original.

Algoritmo 4: Pseudocódigo - LIPNet-PSO-I.

Entrada: Quantidade de camadas 2D L_{2D} , quantidade de camadas 1D L_{1D} e número de partículas n ;
Saída: Melhores parâmetros da LIPNet encontrados: \vec{r}_{best} , $\vec{\sigma}_{best}$, \vec{h}_{best} , $\vec{\sigma}_{best}$, \vec{N}_{best} , $f_{2D,best}$, $f_{1D,best}$;

- 1 Inicializa população com n partículas;
- 2 **enquanto** *condição de parada não for alcançada faça*
- 3 **para cada** *partícula* i **faça**
- 4 **para cada** *camada 2D* l **faça**
- 5 Calcula tamanho mínimo da camada H_{min}^l , usando a Equação (3.2);
- 6 Atualiza $v_i^{r,l}$;
- 7 Atualiza r^l ;
- 8 Atualiza $v_i^{\sigma,l}$;
- 9 Atualiza σ^l ;
- 10 Calcula tamanho da camada H^l , usando a Equação (3.1);
- 11 Atualiza $v_i^{h,l}$;
- 12 Atualiza h^l ;
- 13 Atualiza $v_i^{\sigma,l}$;
- 14 Atualiza σ^l ;
- 15 **para cada** *camada 1D* l **faça**
- 16 Atualiza $v_i^{N,l}$;
- 17 Atualiza N^l ;
- 18 Atualiza $v_i^{f_{2D}}$;
- 19 Atualiza f_{2D} ;
- 20 Atualiza $v_i^{f_{1D}}$;
- 21 Atualiza f_{1D} ;
- 22 Inicializa $cont = 0$ e $media = 0$;
- 23 **enquanto** $cont < q_{fit}$ **faça**
- 24 lipnet.inicializa(\vec{r} , $\vec{\sigma}$, \vec{h} , $\vec{\sigma}$, \vec{N} , f_{2D} , f_{1D});
- 25 lipnet.treina;
- 26 $media = media + lipnet.testa$;
- 27 Incrementa $cont$;
- 28 $fitness_i = media / q_{fit}$;
- 29 Atualiza \vec{p}_i ;
- 30 **para cada** *partícula* i **faça**
- 31 Atualiza \vec{n}_i ;
- 32 Atualiza \vec{n}_{best} ;
- 33 Retorna \vec{n}_{best} .

O tamanho do vetor da partícula t_{part} é calculado segundo a equação:

$$t_{part} = H^0 W^0 + H^{L_{2D}} W^{L_{2D}} + 2 \sum_{l=1}^{L_{2D}-1} (H^l W^l) + N^0 (H^{L_{2D}} W^{L_{2D}} + 1) + \sum_{l=1}^{L_{1D}} N^l (N^{l-1} + 1). \quad (3.3)$$

O Algoritmo 5 sumariza a Arquitetura II.

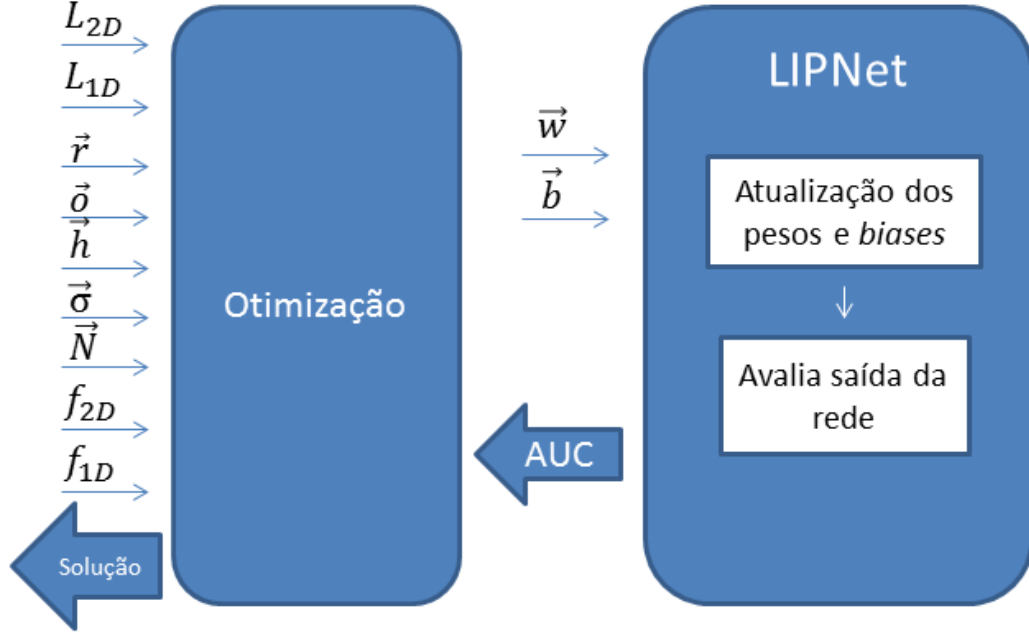


Figura 8 – Relação entre entradas e saídas na Arquitetura II.

Algoritmo 5: Pseudocódigo - LIPNet-PSO-II.

Entrada: L_{2D} , L_{1D} , \vec{r} , \vec{o} , \vec{h} , $\vec{\sigma}$, \vec{N} , f_{2D} , f_{1D} e n ;

Saída: Pesos e polarizações externas de n_{best} ;

- 1 Calcula t_{part} (Eq. (3.3));
 - 2 Inicializa população com n partículas e t_{part} dimensões;
 - 3 **enquanto** condição de parada não for alcançada **faça**
 - 4 **para cada** partícula i **faça**
 - 5 Atualiza \vec{v}_i (Eq. (2.8));
 - 6 Atualiza \vec{x}_i (Eq. (2.9));
 - 7 $lipnet.atualizaPesos(\vec{x}_i)$;
 - 8 $fitness_i = lipnet.avalieRede()$;
 - 9 Atualiza \vec{p}_i ;
 - 10 **para cada** partícula i **faça**
 - 11 Atualiza \vec{n}_i ;
 - 12 Atualiza \vec{n}_{best} ;
 - 13 Retorna n_{best} .
-

4 Experimentos e Resultados

4.1 Introdução

Este capítulo apresenta os resultados obtidos para as soluções propostas no Capítulo 3. Portanto, a Seção 4.2 detalha os experimentos realizados para ambas as arquiteturas. A Seção 4.3 mostra os resultados obtidos para os experimentos definidos.

4.2 Arranjo experimental

Esta seção pretende detalhar os experimentos realizados em ambas as arquiteturas.

Nos dois casos, as imagens foram extraídas a partir do banco de faces do Centro para Aprendizado Biológico e Computacional (CBCL - do inglês, *Center for Biological and Computational Learning*) (HEISELE; POGGIO; PONTIL, 2000). A Figura 9 exemplifica as imagens da base. Ela é dividida em dois bancos: treino e teste. A base de treino é composta por 2429 faces e 4548 não-faces. Já a de teste, 472 faces e 23573 não-faces. As imagens, cujo tamanho é 19×19 , estão em escala de cinza. Para os experimentos, foi aplicada equalização do histograma global. Os bancos de treino e validação empregados nos experimentos foram retirados da base de treino do CBCL. Para os testes, o banco completo de teste da base foi considerado.

Vale ressaltar que ambos os sistemas foram desenvolvidos na linguagem de programação Java. A análise dos resultados se dá pela verificação da AUC e a comparação é feita por meio do teste estatístico de Wilcoxon com 5% de significância.



Figura 9 – Exemplos de imagens do MIT CBCL Face Database: (a) faces de treinamento, (b) faces de teste, (c) não-faces de treinamento e (d) não-faces de teste.[Fonte: extraído de Fernandes (2013)]

4.2.1 Arquitetura I

Os experimentos desta subseção avaliam se o método proposto é capaz de identificar parâmetros da LIPNet que obtenham bons resultados para reconhecimento do padrão em questão.

4.2.1.1 Definição dos experimentos para Arquitetura I

Como o PSO é voltado para ambientes contínuos, é importante que a inicialização das partículas seja analisada a fim de reduzir os riscos do enxame estabilizar em mínimos locais e diminuir o tempo necessário para a convergência. Assim, assume-se que a inicialização da busca precisa ser tendenciosa para obter soluções que caracterizam a LIPNet com efeito de inibição lateral não-nulo, e não a *PyraNet* (inibição lateral nula), pois se presume que a primeira obtém melhores taxas de reconhecimento que a segunda. Dessa maneira, foram definidos 3 experimentos: A, B e C. Os limites máximo e mínimo da inicialização em cada experimento estão destacados na Tabela 3. Para os demais elementos que não estão listados na tabela, o campo de busca da inicialização é o mesmo da otimização. Para todos os experimentos, considerou-se $r_{min} = 2$ e as demais variáveis da Tabela 3 são calculadas conforme as equações destacadas na Tabela 2.

- **Experimento A** - O campo de busca da inicialização é igual ao da otimização.
- **Experimento B** - O campo de busca foi reduzido de modo que a tendência inicial é obter campos receptivos menores e *overlaps* maiores. Desse modo, analisando a Equação (3.1), a média do tamanho da camada 2D é aumentado em relação ao Experimento A. Isto possibilita que o tamanho do campo inibitório possua um maior espaço de busca e, conseqüentemente, maior probabilidade de encontrar soluções com tamanho dos campos inibitórios não-nulos, caracterizando a LIPNet.
- **Experimento C** - A inicialização de r , neste experimento, visa obter tamanhos intermediários para o campo receptivo. Assim, a média de inicialização de r é igual a do Experimento A. Os demais elementos são inicializados tal como no Experimento B.

Tabela 3 – Experimentos para inicialização das partículas na Arquitetura I.

Elemento	A	B	C	
r^l	r_{min} r_{max}	r_{min} $r_{max} - 0,5\Delta r$	$r_{min} + 0,25\Delta r$ $r_{max} - 0,25\Delta r$	Mín Máx
o^l	o_{min} o_{max}	$o_{min} + 0,5\Delta o$ o_{max}	$o_{min} + 0,5\Delta o$ o_{max}	Mín Máx
h^l	h_{min} h_{max}	$h_{min} + 0,5\Delta h$ h_{max}	$h_{min} + 0,5\Delta h$ h_{max}	Mín Máx
σ^l	σ_{min} σ_{max}	σ_{min} $\sigma_{max} - 0,8\Delta\sigma$	σ_{min} $\sigma_{max} - 0,8\Delta\sigma$	Mín Máx

4.2.1.2 Execução dos experimentos para Arquitetura I

Para cada experimento foram realizadas 5 simulações. Quanto à otimização, utilizou-se o critério de parada com 30 iterações, 10 partículas e $q_{fit} = 10$. A fim de comparar com os resultados obtidos por Fernandes (2013), utilizou-se a mesma configuração da LIPNet, ou seja, $L_{2D} = 2$ e $L_{1D} = 1$. O critério de parada do treino de cada rede é por validação cruzada: quando for atingido o limite máximo de 500 épocas, ou a avaliação do banco de validação estagnar por 30 épocas, ou quando o erro da validação crescer por 30 épocas.

Neste caso, ao final do treino, os pesos da rede são atualizados de acordo com os pesos da época na qual o menor erro de validação foi encontrado.

Quanto ao banco positivo de treino, selecionou-se aleatoriamente 100 faces entre as 1000 primeiras da base. O mesmo procedimento ocorreu para o banco negativo de treino. Para o banco de validação, foram consideradas as 500 primeiras imagens não pertencentes ao banco de treino para ambos os bancos positivo e negativo. Vale ressaltar que, uma vez gerados os bancos de treino e validação, eles serão constantes para todas as partículas da simulação em todas as avaliações de *fitness* realizadas.

4.2.2 Arquitetura II

Dois experimentos foram definidos para esta arquitetura. Ambos objetivam comparar a AUC obtida no treinamento da LIPNet original (*Rprop*) com o método proposto na Seção 3.3.

4.2.2.1 Definição dos experimentos para Arquitetura II

As entradas definidas nos Experimentos A e B são aplicadas à rede LIPNet original e à proposta da Arquitetura II para posterior comparação.

- **Experimento A** - Os parâmetros de entrada, que definem a estrutura da LIPNet, são os obtidos em Fernandes, Cavalcanti e Ren (2013), ou seja, $L_{2D} = 2$, $L_{1D} = 1$, $r_1 = 4$, $o_1 = 1$, $r_2 = 4$, $o_2 = 0$, $h_1 = 2$, $\sigma_1 = 1, 6$, $h_2 = 0$. A função de ativação da camada 2D e 1D são, respectivamente, sigmóide logística e tangente hiperbólica.
- **Experimento B** - Os parâmetros de entrada são os obtidos no melhor resultado encontrado entre os experimentos da Arquitetura I.

4.2.2.2 Execução dos experimentos para Arquitetura II

São feitas 10 simulações e a quantidade de partículas é igual a 20. Testou-se dois critérios de parada:

- Validação cruzada: quando for atingido o limite máximo de 2500 iterações, ou a avaliação do banco de validação estagnar por 200 iterações, ou quando o erro da validação crescer por 100 iterações. Ao final da otimização, a solução resultante é a que obteve o menor erro para o banco de validação cruzada em todo o processo;
- Quando for atingido o limite máximo de 2500 iterações.

Quanto ao banco positivo de treino, selecionou-se as 1000 primeiras imagens da base. O mesmo procedimento ocorreu para o banco negativo de treino. Para o banco de validação, foram consideradas 500 imagens aleatórias não pertencentes ao banco de treino para ambos os bancos positivo e negativo.

4.3 Resultados

A Seção 4.3.1 analisa os resultados obtidos para os experimentos da Arquitetura I e a Seção 4.3.2, da Arquitetura II.

4.3.1 Arquitetura I

Como foi descrito na Seção 4.2.1, três tipos de inicialização do enxame são verificadas. Isto visa identificar o melhor espaço para inicialização da busca, visto que o PSO é um algoritmo susceptível a estagnar em mínimos locais e pretende-se reduzir este risco. A Figura 10 mostra os *boxplots* para a melhor AUC encontrada em cada simulação de acordo com cada experimento. A linha tracejada na figura identifica o melhor resultado da LIPNet original.

Entre as cinco soluções encontradas no Experimento A, nenhuma caracterizou uma rede neural LIPNet. O Experimento B está com o *boxplot* mais concentrado, enquanto o Experimento C, mais disperso. Porém, foi este último que identificou a melhor estrutura para a LIPNet com $AUC = 0,89$ dentre todas simulações.

No experimento B, apenas uma das soluções resultou em uma partícula com efeito de inibição lateral não-nulo. Esta mesma solução foi a responsável pelo melhor resultado do experimento. Por fim, no experimento C, quatro das cinco soluções obtiveram estruturas que são específicas da LIPNet, e não, *PyraNet*. Para visualizar com detalhes estes comentários, recorra ao Apêndice A.

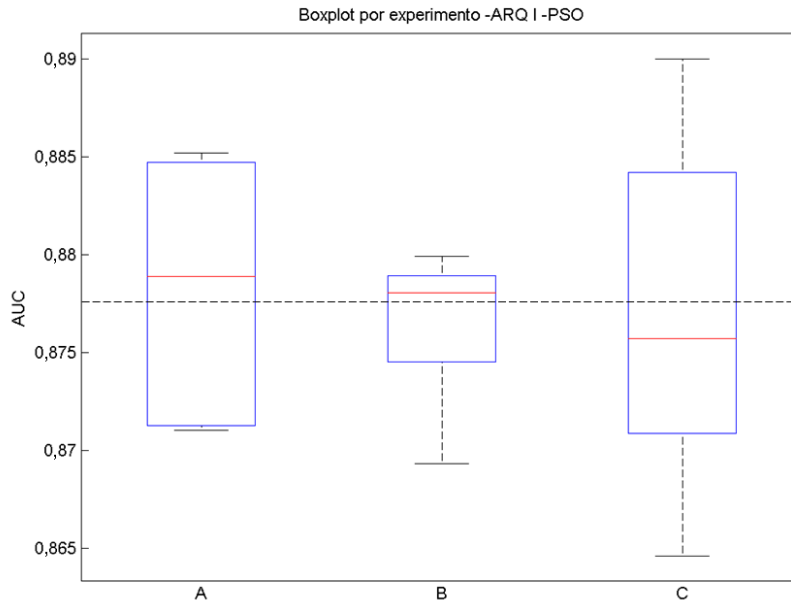


Figura 10 – *Boxplot* da AUC para as melhores soluções encontradas nos Experimentos A, B e C da Arquitetura I. A linha tracejada indica a média da melhor solução encontrada por Fernandes (2013).

A Figura 11 compara a AUC obtida para 10 treinamentos da LIPNet utilizando os parâmetros definidos pela melhor solução encontrada nos experimentos da Arquitetura I (no caso, utilizou-se a solução que obteve $AUC = 0,89$ no Experimento C) e a melhor configuração encontrada por Fernandes (2013). Observa-se que, de fato, a solução encontrada na Arquitetura I é melhor que a do documento original. Ao realizar o teste de Wilcoxon com 5% de significância, tal conclusão é comprovada.

A Figura 12 compara os experimentos através da curva de convergência durante 30 iterações. Para o Experimento A, a curva estabilizou-se nas últimas iterações. Para o Experimento B não houve estagnação.

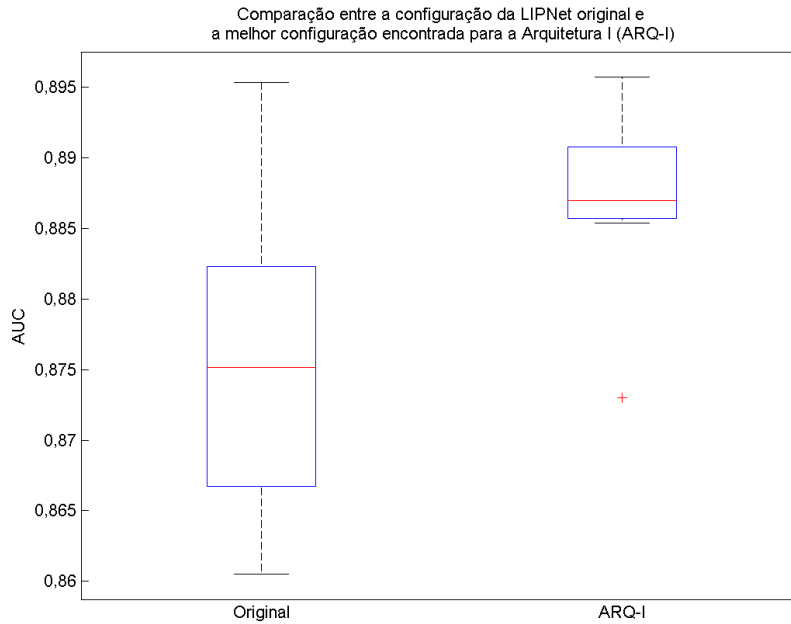


Figura 11 – *Boxplot* que ilustra a AUC obtida para 10 simulações da LIPNet com a configuração da melhor solução entre todos Experimentos da Arquitetura I (ARQ-I) e a melhor solução encontrada por Fernandes (2013)(Original).

Para o Experimento C, a curva aparentou estabilizar por 10 iterações, porém, voltou a convergir no final do processo. Portanto, para uma análise mais minuciosa dos resultados obtidos, é importante aumentar a quantidade de iterações da otimização, porém, essa análise não foi realizada devido ao elevado tempo de execução.

Primeiramente, o modelo foi capaz de encontrar taxas melhores que as encontradas por Fernandes (2013) tanto para a *PyraNet* quanto para a LIPNet. Além disso, a taxa de classificação máxima alcançada corrobora com o resultado obtido em Fernandes (2013), quando o melhor resultado obtido é uma solução com campos inibitórios não-nulos. No entanto, o teste estatístico da soma dos postos de *Wilcoxon* com 5 % de significância concluiu que os três experimentos são estatisticamente equivalentes. Isto permite concluir que, apesar da tentativa de melhorar as taxas de reconhecimento induzindo a inicialização das partículas, o modelo proposto não foi capaz de identificar melhores soluções em alguns casos. Logo, houve convergência para mínimos locais. O PSO é uma técnica específica para espaços de busca contínuos e a maioria das variáveis otimizadas são discretas. Dessa maneira, é possível utilizar outras técnicas mais adequadas ao espaço de busca do problema, tais como algoritmos de computação evolucionária.

4.3.1.1 Considerações sobre o tempo de simulação na Arquitetura I

Naturalmente, um problema enfrentado na Arquitetura I é o tempo de simulação. Considerando um tempo médio t para um treino da LIPNet, o tempo total de uma simulação é igual a:

$$t_{sim} = n_{iteracoes} n_{particulas} q_{fit} t, \quad (4.1)$$

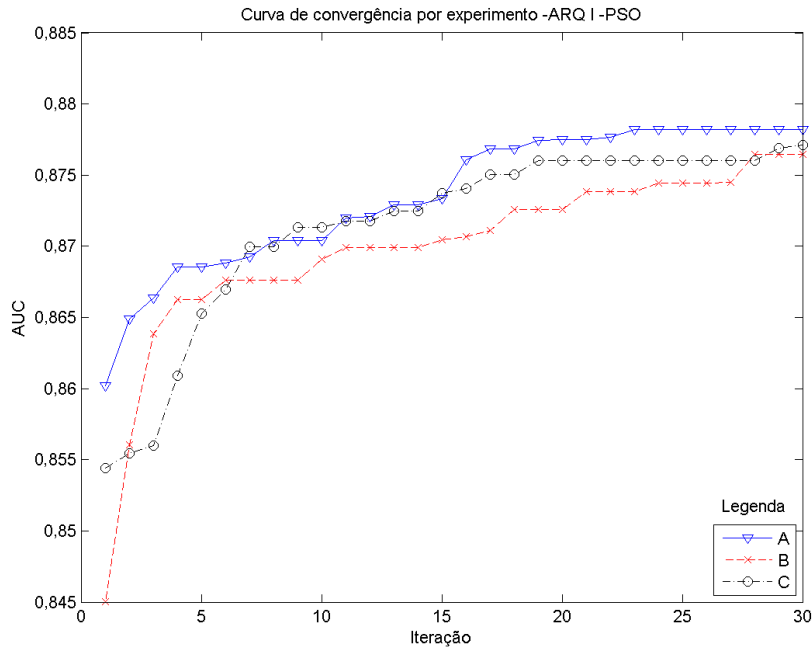


Figura 12 – Comparação da curva de convergência da melhor partícula do PSO entre os Experimentos A, B e C da Arquitetura I.

onde $n_{iteracoes}$ é a quantidade de iterações e $n_{particulas}$ é o total de partículas no enxame.

Para uma simulação do Experimento C, registrou-se a média do tempo de execução de q_{fit} treinamentos realizados em uma avaliação de $fitness$. Esta simulação foi realizada em um computador com 4GB de memória principal, processador Intel Core i5 2,5GHz e sistema operacional Windows 8 de 64 bits. A Figura 13 mostra o *boxplot* com os resultados obtidos em segundos. Nota-se a presença significativa de *outliers*. A ocorrência desses se deve às configurações da LIPNet cujo custo computacional é elevado, nas quais há mais pesos e polarizações externas a serem ajustados durante o treino. A média obtida $t = 10,5284$ segundos, então, o tempo de uma única simulação é $t_{sim} \approx 8,8$ horas, caso seja executado em um sistema serial. A fim de reduzir este tempo de execução, porém, é possível recorrer à paralelização utilizando *clusters* ou CUDA (do inglês, *Compute Unified Device Architecture*), por exemplo.

Apesar do elevado custo computacional, esse modelo é indicado para ser utilizado no início da pesquisa, quando não existe noção sobre quais parâmetros são apropriados para o reconhecimento de certo padrão. O esforço computacional investido inicialmente compensa o tempo gasto caso o ajuste dos parâmetros fosse realizado por meio de tentativa e erro. Assim, este modelo facilita a aplicabilidade da LIPNet em outras bases de imagens.

4.3.2 Arquitetura II

Para o Experimento B, admitiu-se utilizar como entrada para a arquitetura a melhor solução encontrada na Arquitetura I. Tal solução é a que obteve $AUC = 0,89$ no Experimento C. Portanto, conforme destacado no Apêndice A, as configurações de tal estrutura de rede são: $L_{2D} = 2$, $L_{1D} = 1$, $r_1 = 7$, $o_1 = 4$, $r_2 = 4$, $o_2 = 2$, $h_1 = 2$, $\sigma_1 = 5,568$, $h_2 = 0$.

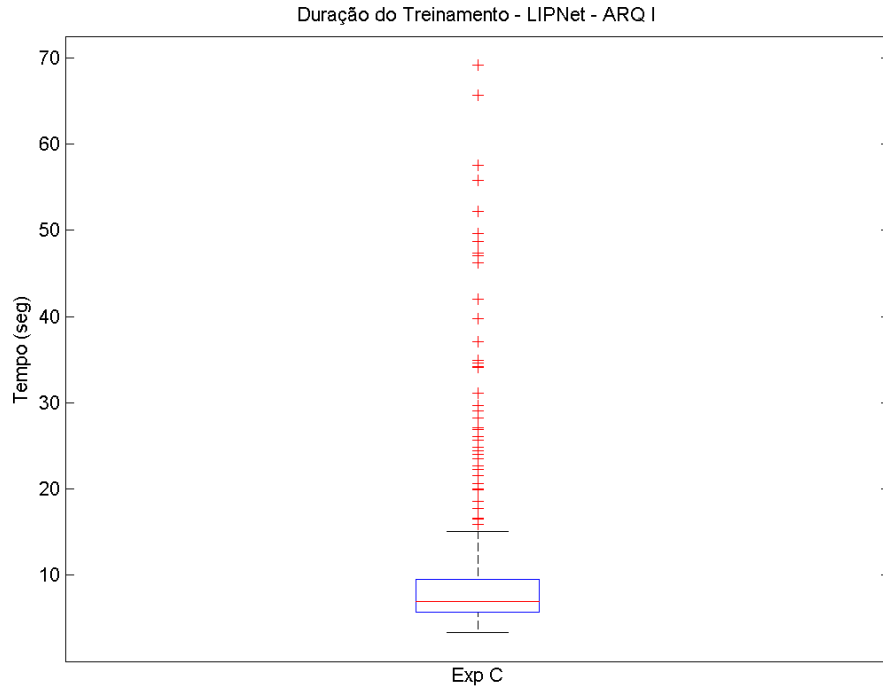


Figura 13 – *Boxplot* do tempo médio de $q_{fit} = 10$ treinamentos em cada avaliação de *fitness* na Arquitetura I.

As Figuras 14 e 15 mostram a AUC obtida para os dois critérios de parada avaliados: Quantidade de Iterações (QI) e Validação Cruzada (VC). Observa-se que a média da AUC obtida em QI é maior que a de VC. O teste da soma dos postos de *Wilcoxon* confirma com 5 % de significância que, de fato, os resultados obtidos por QI são melhores que o de VC. Os critérios de validação cruzada precisam ser reavaliados para um maior número de iterações, visto que os baixos resultados obtidos se devem a uma parada antecipada da simulação.

As Figuras 16 e 17 ilustram a curva de convergência e a de validação cruzada para os experimentos A e B. Ambas curvas ficam estabilizadas por mais de 1000 iterações até acabar o processo de otimização. Esta curva revela que, de fato, a quantidade de iterações igual a 2500 é um ponto ideal para parada da otimização, pois houve estabilidade. Com os parâmetros definidos para VC, a otimização é finalizada prematuramente, acarretando resultados ruins comparativamente a QI. Dessa forma, como trabalhos futuros, é sugerido que seja realizado um estudo para analisar os parâmetros do critério de parada por validação cruzada neste modelo.

Nas Figuras 18 e 19, compara-se o método proposto na Arquitetura II com a método original da LIPNet (*RProp*) para os experimentos A e B, respectivamente. Realizando o teste de *Wilcoxon* com 5% de significância, conclui-se que a LIPNet obteve melhores resultados. Isto se deve porque o PSO, como já citado anteriormente, é susceptível a convergir para mínimos locais. Portanto, é necessário recorrer a outras técnicas mais robustas a este tipo de problema, a fim de obter melhores taxas de reconhecimento.

As Figuras 20 e 21 ilustram a curva ROC para os experimentos A e B da Arquitetura II, representada

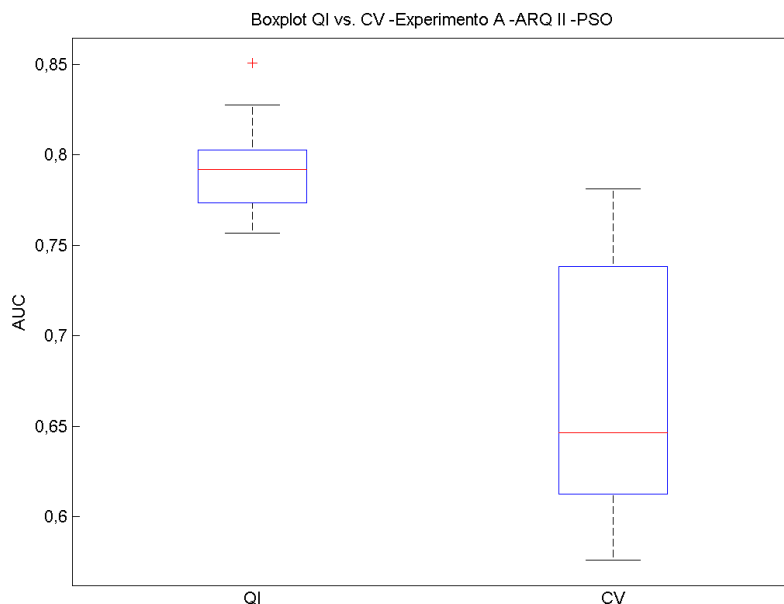


Figura 14 – *Boxplot* da AUC obtida para dois critérios de parada da otimização da Arquitetura II no Experimento A: Quantidade de Iterações (QI) e Validação Cruzada (VC).

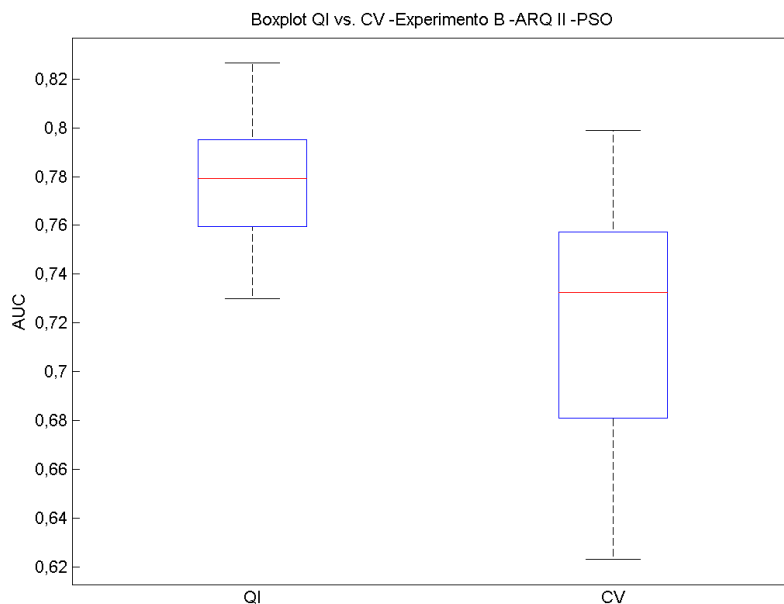


Figura 15 – *Boxplot* da AUC obtida para dois critérios de parada da otimização da Arquitetura II no Experimento B: Quantidade de Iterações (QI) e Validação Cruzada (VC).

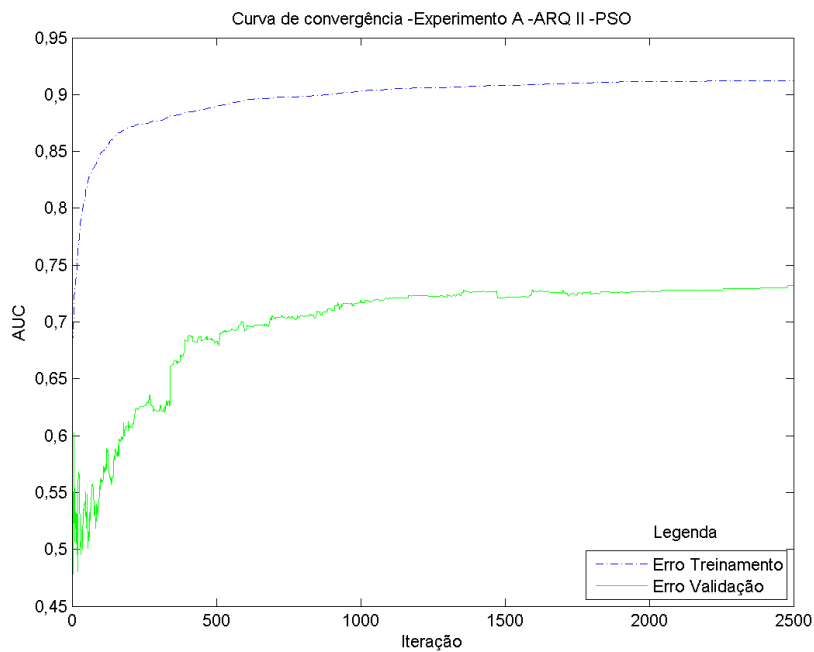


Figura 16 – Curva de convergência versus validação cruzada para o Experimento A da Arquitetura II.

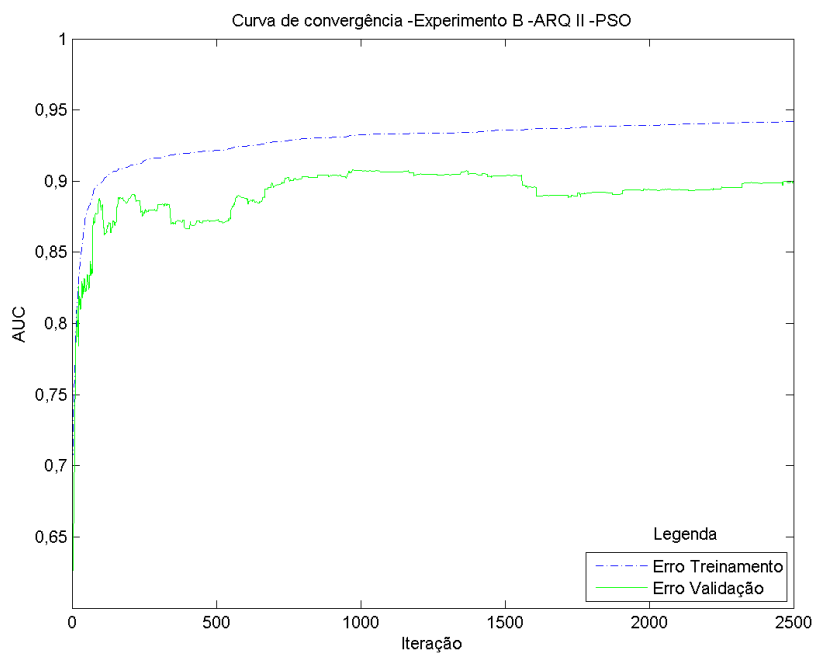


Figura 17 – Curva de convergência versus validação cruzada para o Experimento B da Arquitetura II.

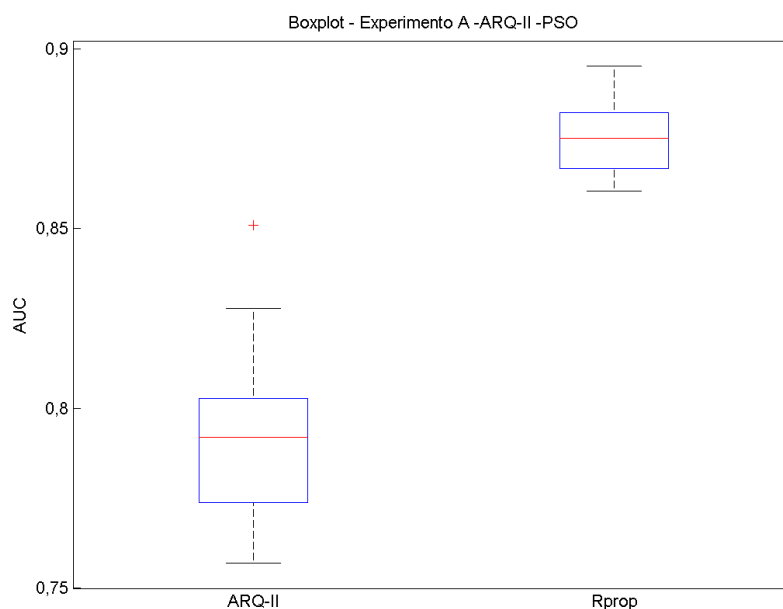


Figura 18 – *Boxplot* da AUC que compara o resultado obtido no Experimento A com a atualização dos pesos na proposta da Arquitetura II (ARQ-II) com o método original da LIPNet (*Rprop*).

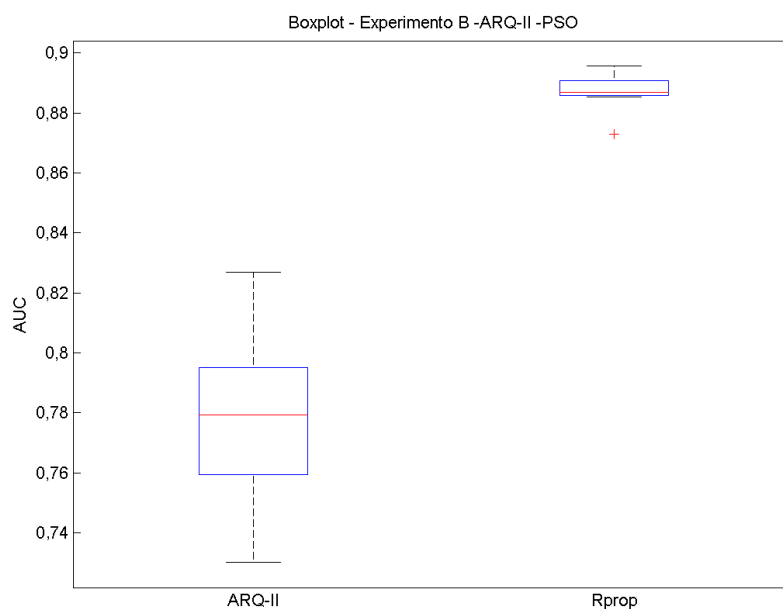


Figura 19 – *Boxplot* da AUC que compara o resultado obtido no Experimento B com a atualização dos pesos na proposta da Arquitetura II (ARQ-II) com o método original da LIPNet (*Rprop*).

com os círculos, e para a LIPNet original, com os asteriscos.

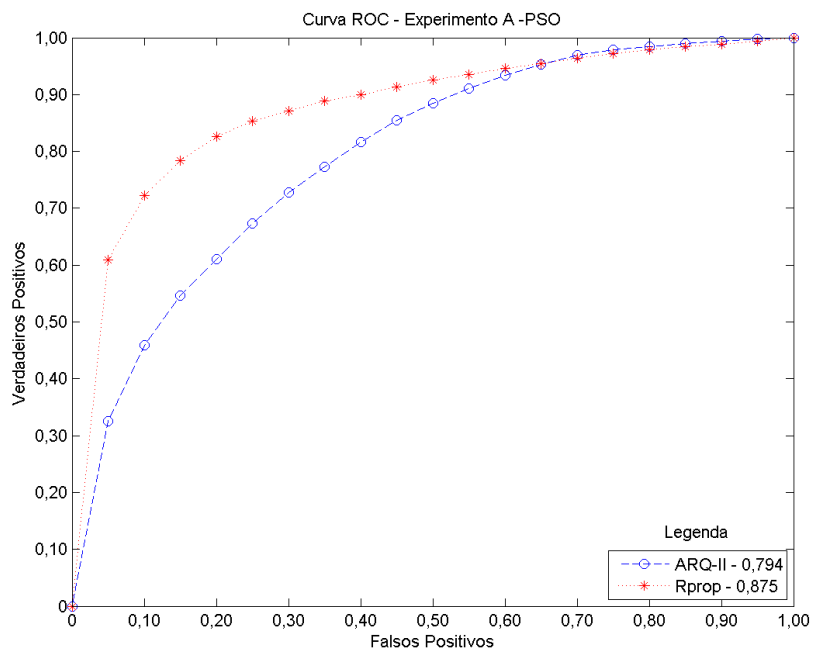


Figura 20 – Curva ROC do Experimento A da Arquitetura II.

Pode-se concluir que, para esta Arquitetura, os resultados obtidos foram similares em ambos os experimentos.

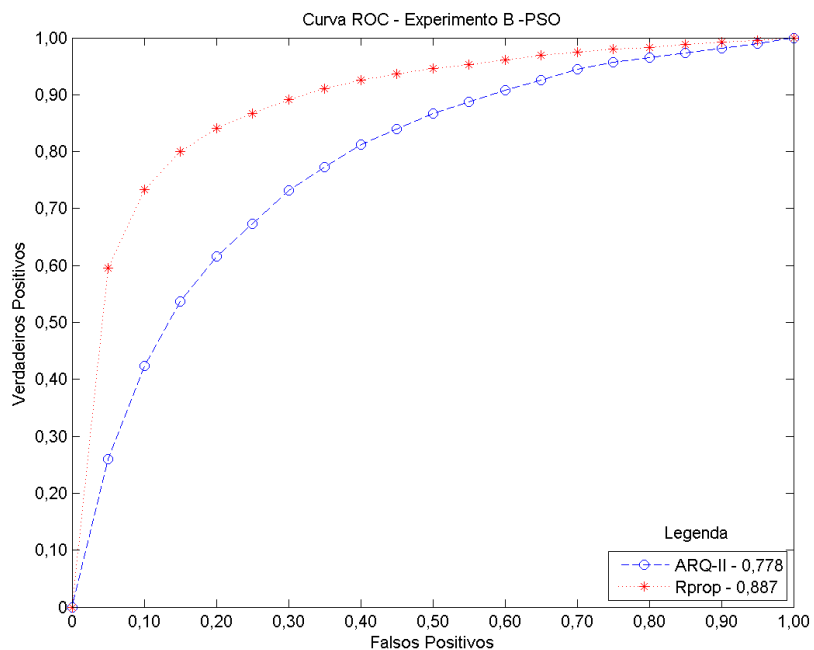


Figura 21 – Curva ROC do Experimento B da Arquitetura II.

5 Conclusões e Trabalhos Futuros

Neste capítulo serão apresentadas as conclusões observadas durante o desenvolvimento deste trabalho, bem como os trabalhos futuros, que poderão ser realizados em complementação deste.

5.1 Conclusões

A LIPNet é uma rede neural inspirada nos conceitos de campos receptivos e inibitórios presentes no sistema visual humano. Nesta rede, a técnica utilizada para o treinamento dos pesos, *Rprop*, é vulnerável a mínimos locais. Este trabalho propôs duas soluções que empregam o PSO, algoritmo de inteligência de enxames, a fim de contornar esse problema. Porém, outras técnicas de otimização podem substituí-lo nesta função.

A Arquitetura I proposta visa a otimização dos parâmetros da rede, tais como, tamanho dos campos receptivos, inibitórios e função de ativação. Este trabalho analisou diferentes formas de inicialização, a fim de tender os resultados para configurações da LIPNet (tamanho dos campos inibitórios não-nulo). O modelo identificou taxas melhores que as encontradas por Fernandes (2013) tanto para a *PyraNet* quanto para a LIPNet. Além disso, a taxa de classificação máxima alcançada corrobora com o resultado obtido por Fernandes (2013), quando o melhor resultado obtido é uma solução com campos inibitórios não-nulos. No entanto, verificou-se que os resultados obtidos são estatisticamente equivalentes a formas de inicialização inclinadas para a rede *PyraNet* (tamanho de campos inibitórios nulo). Isto indica que o modelo ainda não é robusto o suficiente a mínimos locais. É recomendável que este modelo seja utilizado no início da pesquisa, quando não existe noção sobre quais parâmetros são apropriados para o reconhecimento de certo padrão. Sua desvantagem, em contrapartida, é o custo computacional elevado para a otimização.

O segundo modelo é a Arquitetura II, a qual objetiva substituir a técnica *Rprop* de ajuste dos pesos pela otimização com o PSO. Dessa forma, o treino da rede consiste no próprio processo de otimização, no qual os pesos da rede são as variáveis do espaço de busca. Concluiu-se que o PSO não é uma técnica ideal para este modelo, pois ele foi mais vulnerável a mínimos locais que o próprio *Rprop*, resultando em taxas de reconhecimento baixas comparativamente à LIPNet original. Portanto, é interessante substituir o PSO por outra técnica de otimização mais adequada a este problema.

5.2 Trabalhos Futuros

Como trabalhos futuros, destacam-se:

1. Em relação à Arquitetura I, propor modelo híbrido com técnicas computacionais apropriadas para ambientes discretos, tais como, computação evolucionária;
2. Em relação à Arquitetura II, substituir PSO por outras técnicas menos vulneráveis a mínimos locais, tais como o FSS-II (do inglês, *Fish School Search II*) (BASTOS-FILHO; NASCIMENTO, 2013);
3. Em relação à Arquitetura II, avaliar critérios de parada para a otimização, como, por exemplo, variar os parâmetros de validação cruzada;
4. Estender a LIPNet para um modelo com campos receptivos e inibitórios de tamanhos variáveis;
5. Testar os modelos propostos para outras bases, a fim de analisar a profundidade da rede.

Referências

- Bastos-Filho, C. J. A. and Nascimento, D. O. An Enhanced Fish School Search Algorithm. In: *BRICS Conference on Computational Intelligence*. [S.l.: s.n.], 2013.
- C. M. Bishop. *Neural Networks for Pattern Recognition*. [S.l.]: Clarendon, 2007.
- A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recognition*, v. 30, n. 7, p. 1145–1159, 1997.
- Bratton, Daniel and Kennedy, James. Defining a Standard for Particle Swarm Optimization. *IEEE Swarm Intelligence Symposium*, n. Sis, p. 120–127, 2007.
- R. Coultrip and R. Granger and G. Lynch. A cortical model of winner-take-all competition via lateral inhibition. *Neural Networks*, v. 5, n. 1, p. 47–54, 1992.
- Fernandes, Bruno José Torres. *Redes Neurais com Extração Implícita de Características para Reconhecimento de Padrões Visuais*. Tese (Doutorado) — Universidade Federal de Pernambuco, 2013.
- Bruno José Torres Fernandes and Cavalcanti, George D C and Ren, Tsang Ing. Lateral inhibition pyramidal neural network for image classification. *Systems, Man and Cybernetics, IEEE Transactions*, PP, p. 2168–2267, 2013.
- Fukushima, K. and Miyake, S. and Ito, T. Neocognitron: A neural network model for a mechanism of visual pattern recognition. *Systems, man and Cybernetics, IEEE Transactions*, SMC-13, n. 5, p. 826–834, 1983.
- Ganis, M. D. and Wilson, C. L. and Blue, J. L. Neural network-based systems for handprint ocr applications. *Image Processing, IEEE Transactions*, v. 7, n. 8, p. 1097–1112, 1998.
- Rafael C. Gonzalez and Richard E. Woods. *Processamentos Digital de Imagens*. [S.l.]: Prentice-Hall, 2010.
- H. K. Hartline. The response of single optic nerve fibers of the vertebrate eye to illumination of the retina. *American Journal of Phisiology*, 1938.
- Heisele, B. and T. Poggio and M. Pontil. *Face detection in still gray images*. [S.l.], 2000.
- L. Hermann. Eine erscheinung simultanen contrastes. *Pflügers Archiv für die gesamte Physiologie*, v. 3, p. 13–15, 1870.
- D. H. Hubel. The visual cortex of the brain. *Scientific American*, p. 54–62, 1963.
- J. Kennedy and R. C. Eberhart. Particle swarm optimization. In *Proc. IEEE Int. Conference on Neural Networks*, p. 1942–1948, 1995.
- LeCun, Y. and Boser, B. and Denker, J. S. and Henderson, D. and Howard, R. E. and Hubbard, W. and Jackel, L. D. Backpropagation applied to handwritten zip code recognition. *Neural Comput.*, v. 1, n. 4, p. 541–551, 1989.
- LeCun, Y. and Bottou, L. and Bengio, Y. and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings IEEE*, v. 86, n. 11, p. 2278–2324, 1998.
- M. Levine and J. Shefner. *Fundamentals of sensation and perception*. [S.l.]: Oxford Univ. Press, 2000.
- Mao, Z.-H., and Massaquoi, S. G. Dynamics of winner-take-all competition in recurrent neural networks with lateral inhibition. *Neural Networks, IEEE Transactions*, v. 18, n. 1, p. 55–69, 2007.

- C. A. Perez and C. A. Salinas and P. A. Estévez and P. M. Valenzuela. Genetic design of biologically inspired receptive fields for neural pattern recognition. *IEEE Transactions on Systems, Man and Cybernetics*, v. 33, n. 2, p. 258–270, 2003.
- Phung, S. L. and Bouzerdoum, A. A pyramidal neural network for visual pattern recognition. *Neural Networks, IEEE Transactions*, v. 18, n. 2, p. 329–343, 2007.
- M. Riedmiller and H. Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *Proc. IEEE Int. Conf. Neural Networks*, p. 586–591, 1993.
- G. Rizzolatti and R. Camarda. Inhibition of visual responses of single units in the cat visual area of the lateral suprasylvian gyrus (Clare-Bishop area) by the introduction of a second visual stimulus. *Brain Research*, v. 88, n. 2, p. 357–361, 1975.
- H. A. Rowley and S. Baluja and T. Kanade. Neural network-based face detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions*, v. 20, n. 1, p. 23–38, 1998.
- Santos, S. M. O. C. and Valença, M. J. S. V. and Bastos-Filho, C. J. A. B. Comparing particle swarm optimization approaches for training multi-layer perceptron neural networks for forecasting. *Lecture Notes in Computer Science*, v. 7435, p. 344–351, 2012.
- Teixeira, L. A. and Toledo F. S. R. and Oliveira, A. L. I. and Bastos-Filho, C. J. A. Adjusting weights and architecture of neural networks through pso with time-varying parameters and early stopping. In: *SBRN'2008 (the 10th Brazilian Symposium on Neural Networks)*. [S.l.: s.n.], 2008. p. 33–38.
- XuGang Wang and Hongan Wang and Guozhong Dai and Zheng Tang. A reliable resilient backpropagation method with gradient ascent. In: *ICIC (2)*. [S.l.: s.n.], 2006. p. 236–244.

APÊNDICE A – Soluções encontradas nas simulações da Arquitetura I

Este apêndice visa ilustrar os resultados obtidos para as simulações dos experimentos da Arquitetura I. As Figuras 22, 23 e 24 destacam as cinco soluções encontradas provenientes desses experimentos. Para cada solução é destacado:

- quantidade de camadas 2D, representada por '*quantidadeCamada2D*'. Vale destacar que o valor destacado na figura equivale a $L_{2D} + 1$, pois conta com a camada de entrada;
- quantidade de camadas 1D, representada por '*quantidadeCamada1D*';
- tamanho de cada camada 2D, representado por '*tamanhoCamada2D*'. Neste caso, são apresentados três valores pois correspondem ao tamanho da primeira, segunda e terceira camadas respectivamente;
- tamanho do campo receptivo, representado por '*tamanhoCampoReceptivo*'. Neste caso, são apresentados três valores pois correspondem ao tamanho dos campos receptivos na primeira, segunda e terceira camadas respectivamente;
- tamanho do *overlap*, representado por '*tamanhoOverlap*'. Neste caso, são apresentados três valores pois correspondem ao tamanho do *overlap* na primeira, segunda e terceira camadas respectivamente;
- tamanho do campo inibitório, representado por '*tamanhoCampoInibitorio*'. Neste caso, são apresentados três valores pois correspondem ao tamanho dos campos inibitórios na primeira, segunda e terceira camadas respectivamente;
- peso inibitório, representado por '*pesoInibitorio*'. Neste caso, são apresentados três valores pois correspondem ao peso inibitório na primeira, segunda e terceira camadas respectivamente. Vale ressaltar que, caso o tamanho do campo inibitório na camada seja nulo, não importa a magnitude do peso, pois não haverá efeito de inibição lateral na respectiva camada;
- quantidade de neurônio na camada 1D, representado por '*tamanhoCamada1D*';
- função de ativação dos neurônios da camada 2D e 1D, representados respectivamente por '*funcaoAtivacaoCamada2D*' e '*funcaoAtivacaoCamada1D*'. Neste caso, os valores assumidos são '*SigmLog*', indicando a função sigmóide logística, e '*TangHip*', a tangente hiperbólica;
- avaliação de *fitness* obtida para os referidos parâmetros é representada por '*Fitness*'.

A.1 Experimento A da Arquitetura I

As soluções do Experimento A podem ser visualizadas na Figura 22. Ao verificar as cinco simulações, percebe-se que o tamanho do campo inibitório obtido em todos os casos é nulo, o que caracteriza uma rede *PyraNet*.

A.2 Experimento B da Arquitetura I

As soluções do Experimento B podem ser visualizadas na Figura 23. Ao verificar as cinco simulações, percebe-se que o tamanho do campo inibitório obtido não é nulo apenas na Solução 3. Essa solução inclusive foi a que obteve segundo melhor *fitness*, o que evidencia que a inibição lateral é capaz de obter taxas de reconhecimento tão boas quanto a *PyraNet*.

A.3 Experimento C da Arquitetura I

As soluções do Experimento C podem ser visualizadas na Figura 24. Ao verificar as cinco simulações, percebe-se que o tamanho do campo inibitório obtido é nulo apenas na Solução 3. Vale ressaltar portanto que o modo de inicialização das variáveis no espaço de busca foi essencial para obter uma maior quantidade de soluções que atendam às configurações da LIPNet com inibição lateral não-nula.

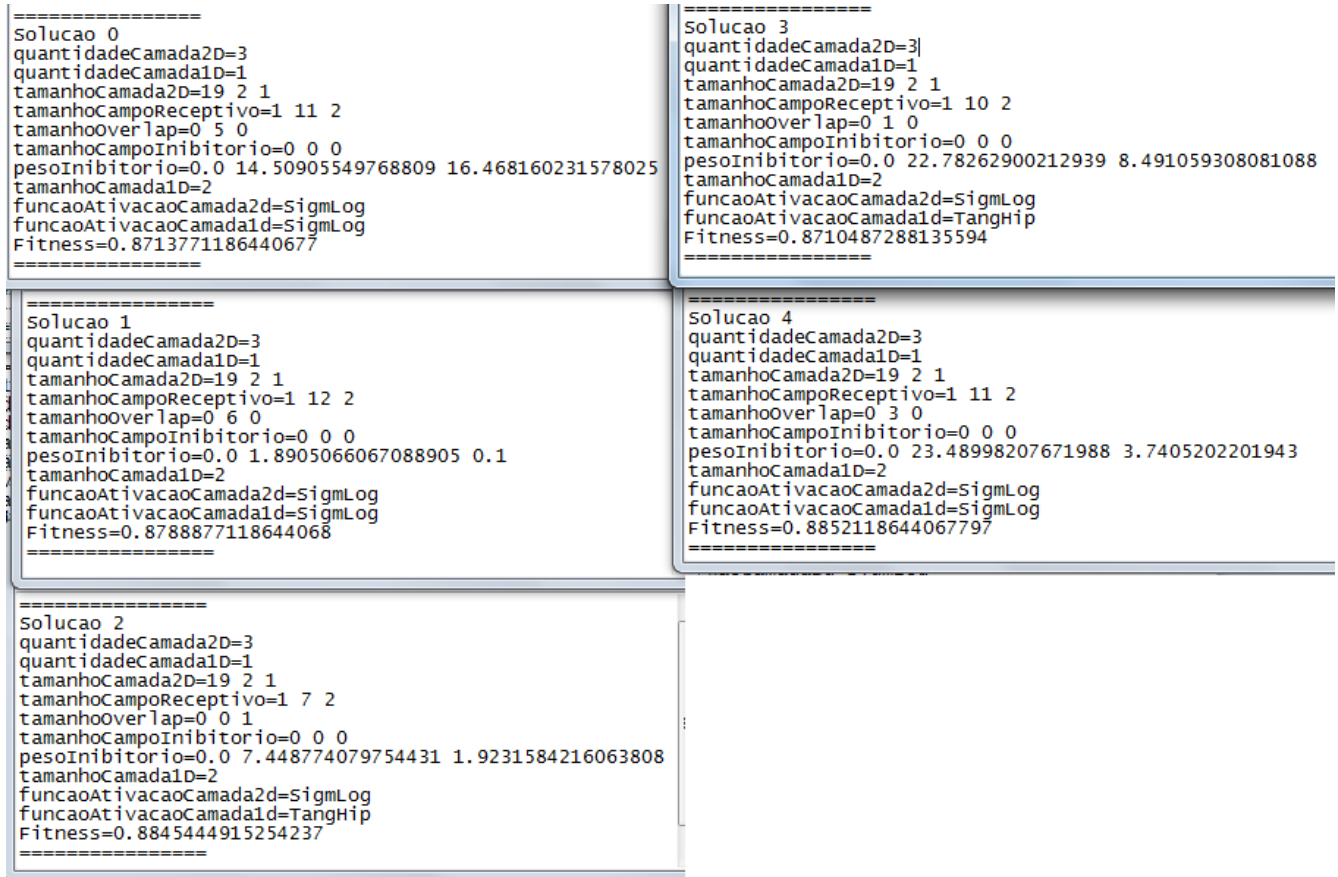


Figura 22 – Soluções encontradas para o Experimento A - Arquitetura I.


```

=====
Solucao 0
quantidadeCamada2D=3
quantidadeCamada1D=1
tamanhoCamada2D=19 2 1
tamanhoCampoReceptivo=1 8 2
tamanhoOverlap=0 2 1
tamanhoCampoInibitorio=0 0 0
pesoInibitorio=0.0 9.735611176441783 1.014120862179213
tamanhoCamada1D=2
funcaoAtivacaoCamada2d=SigMLog
funcaoAtivacaoCamada1d=SigMLog
Fitness=0.8799470338983051
=====

=====
Solucao 2
quantidadeCamada2D=3
quantidadeCamada1D=1
tamanhoCamada2D=19 2 1
tamanhoCampoReceptivo=1 7 2
tamanhoOverlap=0 0 1
tamanhoCampoInibitorio=0 0 0
pesoInibitorio=0.0 9.101926173368586 30.0
tamanhoCamada1D=2
funcaoAtivacaoCamada2d=SigMLog
funcaoAtivacaoCamada1d=TangHip
Fitness=0.8762500000000001
=====

=====
Solucao 1
quantidadeCamada2D=3
quantidadeCamada1D=1
tamanhoCamada2D=19 2 1
tamanhoCampoReceptivo=1 10 2
tamanhoOverlap=0 3 1
tamanhoCampoInibitorio=0 0 0
pesoInibitorio=0.0 22.702818470544987
26.071039357587274
tamanhoCamada1D=2
funcaoAtivacaoCamada2d=SigMLog
funcaoAtivacaoCamada1d=SigMLog
Fitness=0.869343220338983
=====

=====
Solucao 3
quantidadeCamada2D=3
quantidadeCamada1D=1
tamanhoCamada2D=19 6 1
tamanhoCampoReceptivo=1 4 4
tamanhoOverlap=0 1 0
tamanhoCampoInibitorio=0 2 0
pesoInibitorio=0.0 7.702246535203959 2.262184921103894
tamanhoCamada1D=2
funcaoAtivacaoCamada2d=SigMLog
funcaoAtivacaoCamada1d=TangHip
Fitness=0.8785911016949152
=====

=====
Solucao 4
quantidadeCamada2D=3
quantidadeCamada1D=1
tamanhoCamada2D=19 5 1
tamanhoCampoReceptivo=1 6 4
tamanhoOverlap=0 3 1
tamanhoCampoInibitorio=0 0 0
pesoInibitorio=0.0 0.1 28.904477394152718
tamanhoCamada1D=2
funcaoAtivacaoCamada2d=sigMLog
funcaoAtivacaoCamada1d=TangHip
Fitness=0.8780720338983052
=====
    
```

Figura 23 – Soluções encontradas para o Experimento B - Arquitetura I.

```

=====
Solucao 0
quantidadeCamada2D=3
quantidadeCamada1D=1
tamanhoCamada2D=19 6 1
tamanhoCampoReceptivo=1 8 6
tamanhoOverlap=0 6 5
tamanhoCampoInibitorio=0 2 0
pesoInibitorio=0.0 7.19019084337727 13.628156609291072
tamanhoCamada1D=2
funcaoAtivacaoCamada2d=SigMLog
funcaoAtivacaoCamada1d=SigMLog
Fitness=0.8645974576271186
=====

=====
Solucao 3
quantidadeCamada2D=3
quantidadeCamada1D=1
tamanhoCamada2D=19 5 1
tamanhoCampoReceptivo=1 7 4 |
tamanhoOverlap=0 4 1
tamanhoCampoInibitorio=0 0 0
pesoInibitorio=0.0 0.1 26.066759637947243
tamanhoCamada1D=2
funcaoAtivacaoCamada2d=SigMLog
funcaoAtivacaoCamada1d=TangHip
Fitness=0.8822775423728814
=====

=====
Solucao 1
quantidadeCamada2D=3
quantidadeCamada1D=1
tamanhoCamada2D=19 5 1
tamanhoCampoReceptivo=1 7 4
tamanhoOverlap=0 4 0
tamanhoCampoInibitorio=0 2 0
pesoInibitorio=0.0 0.1 1.728156777904342
tamanhoCamada1D=2
funcaoAtivacaoCamada2d=SigMLog
funcaoAtivacaoCamada1d=TangHip
Fitness=0.8757415254237289
=====

=====
Solucao 4
quantidadeCamada2D=3
quantidadeCamada1D=1
tamanhoCamada2D=19 5 1
tamanhoCampoReceptivo=1 7 4
tamanhoOverlap=0 4 2
tamanhoCampoInibitorio=0 2 0
pesoInibitorio=0.0 5.567708719098397 5.473451344629539
tamanhoCamada1D=2
funcaoAtivacaoCamada2d=SigMLog
funcaoAtivacaoCamada1d=SigMLog
Fitness=0.8900211864406782
=====

=====
Solucao 2
quantidadeCamada2D=3
quantidadeCamada1D=1
tamanhoCamada2D=19 12 3
tamanhoCampoReceptivo=1 8 4
tamanhoOverlap=0 7 1
tamanhoCampoInibitorio=0 2 1
pesoInibitorio=0.0 0.1 13.216781595387179
tamanhoCamada1D=2
funcaoAtivacaoCamada2d=SigMLog
funcaoAtivacaoCamada1d=SigMLog
Fitness=0.8729661016949153
=====
    
```

Figura 24 – Soluções encontradas para o Experimento C - Arquitetura I.