

**UNIVERSIDADE ESTADUAL DE CAMPINAS**

**MS777 - Projeto Supervisionado -  
Redes Neurais Morfológicas em Análise de Imagens  
Prof. Estevão Esmi Laureano**

Vinícius Figueiredo Fernandes (157510)

**Sumário**

<b>1</b>	<b>Introdução ao projeto</b>	<b>1</b>
<b>2</b>	<b>Introdução às Redes Neurais</b>	<b>2</b>
<b>3</b>	<b>Conceitos de Álgebra de Imagem</b>	<b>5</b>
<b>4</b>	<b>Noções de Morfologia Matemática</b>	<b>8</b>
<b>5</b>	<b>As Redes Neurais Morfológicas</b>	<b>10</b>

**Campinas  
Junho de 2019**

## 1 Introdução ao projeto

A área de inteligência computacional é relativamente recente, porém, já acumula grandes conquistas. E isso é ainda mais evidente para o caso do *Deep Learning*, uma de suas subáreas, que possui amplo espaço para especulação e pesquisa, como pôde ser observado em 2016, quando o programa de *Deep learning* Alpha Go ganhou o prêmio da revista Science como Breakthrough of the Year.[15] Essa subárea recebe grande atenção da comunidade científica, em que todos os anos novos modelos são criados para serem integrados, adaptados e modificados.

Além disso, a incorporação da álgebra de imagens e morfologia matemática na teoria das redes neurais, ocasionou em uma ramificação das redes neurais artificiais clássicas, surgindo as redes neurais morfológicas, um tópico ainda mais recente e pouco explorado na grande área da inteligência computacional, na qual as redes neurais fazem parte.

Este projeto tem como tarefa investigar conceitos e resultados das amplas áreas de álgebra de imagens, morfologia matemática e redes neurais, necessárias para a compreensão do que seriam as redes neurais artificias morfológicas.

## 2 Introdução às Redes Neurais

A área de inteligência computacional pode ser dita tão antiga quanto a própria área da computação. Alan Turing, o pai da computação, em 1950 já havia publicado um artigo pioneiro[1] questionando a capacidade das máquinas pensarem. Em 1956, dois anos após a morte de Turing, John MacCarthy cunhou o termo inteligência artificial. E por fim, em 1994, Bedzek definiu pela primeira vez o termo inteligência computacional como sendo um subconjunto da pesquisa em inteligência artificial com objetivos a curto prazo, tal como resolver tarefas simples da engenharia com escopos bem delineados, sem focar em questões gerais e abstratas referentes ao sentido amplo de inteligência. Esse subcampo de pesquisa inclui as famosas redes neurais artificiais e os sistemas de lógica fuzzy.

Além disso, temos como paradigma da inteligência computacional, a aprendizagem de máquina (*machine learning*) que consiste na capacidade dos sistemas computacionais de adquirirem conhecimento extraíndo padrões de uma base de dados. A aprendizagem pode ser ainda dividida em não-supervisionada, supervisionada e aprendizagem por reforço, dependendo da estrutura da base de dados fornecida à máquina e ao algoritmo utilizado para a extração dos padrões.

As redes neurais artificiais que são modelos computacionais da inteligência computacional que podem ser usados com qualquer um dos paradigmas de aprendizagem de máquina, porém, ao longo desse projeto, nos preocuparemos com as redes neurais que utilizam-se da aprendizagem supervisionada.

A unidade básica de uma rede neural artificial é o perceptron, um modelo que recebe diversas entradas binárias e fornece como saída um único valor binário. A saída é definida por uma soma entre os pesos, entradas e *biases* (equação 1). E possui como cerne de seu estudo a determinação dos pesos ( $w$ ) e *biases* ( $b$ ).

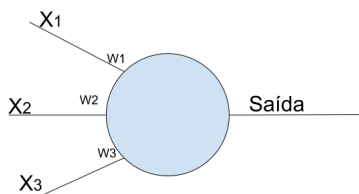


Figura 1: Modelo simples de um perceptron

$$y = \begin{cases} 1, & \text{se } \sum_{i=1}^n w_i x_i + b \geq 0 \\ 0, & \text{caso contrário} \end{cases} \quad (1)$$

Esse modelo foi desenvolvido nas décadas de 50 e 60 pelo cientista Frank Rosenblatt [3], inspirado pelos trabalhos de Warren McCulloch e Walter Pitts[4]. A ideia central para essa unidade foi bioinspirada pela teoria neurofisiológica desses dois cientistas, em que estudaram eventos neurológicos através de modelos

de lógica proposicional. Nesse contexto, um perceptron corresponderia a um neurônio.

O entendimento atual de como esse modelo aprende seus pesos e *biases* é fortemente baseado em heurística e portanto, a criação e melhoramento de algoritmos e modificações de modelos também são construídos heurísticamente. Infelizmente, um único perceptron não é capaz de resolver com grande eficiência muitos problemas computacionais, portanto, necessitamos criar modelos que sustentem mais de um neurônio.

Um exemplo clássico da limitação do perceptron é o problema de classificação na criação de uma porta XOR (exclusive or). Para a criação da porta XOR, uma rede recebe 2 entradas binárias e retorna a saída esperada para essa porta lógica, como pode ser visto na tabela abaixo. Esse problema, no entanto, não é linearmente separável, isto é, não é possível com uma única reta separar as classes de maneira apropriada. Isso torna o problema da porta XOR impossível de ser resolvido por um único perceptron.[10]

$A$	$B$	$A \text{ XOR } B$
0	0	0
0	1	1
1	0	1
1	1	0

Tabela 1: Porta XOR

Por outro lado, é possível demonstrar que um único perceptron consegue formar uma porta NAND, com escolhas apropriadas de pesos e *biases*. E como essas portas são universais para a computação, perceptrons são universais para a computação.[7] Isto significa que utilizando de perceptrons como unidades básicas, é possível realizar qualquer operação computacional, portanto, sendo possível a criação de uma porta XOR. Uma arquitetura clássica de rede neural artificial de múltiplas camadas, consiste em uma camada de entradas, múltiplas camadas escondidas e uma última camada de saída.

Um segundo melhoramento para as redes neurais artificiais é a mudança de um sistema binário de saída para um sistema de saída contínua em um intervalo. Para isso, utiliza-se uma função de ativação sobre as entradas, ao invés do sistema de regras simples anterior.

$$y = \sigma(wx + b), \tag{2}$$

em que para o propósito deste trabalho, vamos assumir que a função  $\sigma$  é continuamente diferenciável.

*Deep learning* é mais um dos conceitos pertinentes à ampla área de inteligência computacional e surge do aprofundamento dos estudos na área de redes neurais artificiais. Costuma-se classificar um modelo como um modelo *deep*, se esse envolve uma rede neural artificial com várias camadas.[7]

Um modelo *deep* consegue entender os conceitos abstratos associados aos dados independente de seus fatores de variações. Por exemplo, através do *Deep learning* o sistema consegue reconhecer um carro independentemente da luminosidade da imagem. Heuristicamente entende-se que isso ocorre pois os conceitos de formas geométricas do objeto “carro” são abstraídos de contexto. Dessa maneira, uma rede neural artificial *deep* permite ao computador aprender pela sua própria experiência, construindo uma hierarquia própria de conhecimento e formação própria de abstração.

Contudo, no estudo de um único perceptron ou de uma rede neural artificial *deep* o cerne da questão é o mesmo, encontrar bons pesos e *biases*. No caso da grande maioria das redes neurais de múltiplas camadas, encontrar os pesos e *biases* apropriados é um problema NP-completo[13], por isso é tão fundamental criar um algoritmo que contorne uma formulação analítica. Assim surgem os diversos algoritmos para a identificação dos parâmetros, sendo o mais conhecido desses o algoritmo de Backpropagation.[14]

Encontrar pesos e *biases* apropriados não é trivial, a ideia seria então, definir uma função custo, denotada por  $C$ , que avalia o quão próxima a saída produzida da rede está de uma saída ou resposta desejada, e através das pequenas variações dessa, isto é, através de suas derivadas parciais, poderíamos encontrar novos valores para os pesos e os *biases* que minimizem  $C$ .

Note que aqui, apenas podemos definir a função custo como função da saída real e esperada, pois, estamos tratando de uma aprendizagem de máquina supervisionada e, portanto, para os problemas de classificações, já conhecemos os rótulos esperados para cada entrada.

O algoritmo de Backpropagation conjuntamente com o gradiente descendente estocástico nos permite computar as derivadas parciais da função custo em função do peso e do *bias* com respeito a qualquer peso e *bias* da rede. Isto significa que não importa em qual camada estivermos, podemos calcular a mudança do peso das camadas anteriores com base na mudança do peso na camada atual, por isso o nome backpropagation. Neste algoritmo, as mudanças nos parâmetros ocorrem da última camada para a primeira.

### 3 Conceitos de Álgebra de Imagem

Álgebra de imagem é uma álgebra multi-valorada ou heterogênea no sentido de Birkhoff e Lipson, possuindo múltiplos conjuntos de operandos e operadores. A álgebra de imagens surge como ferramenta matemática rigorosa no estudo de processamentos de imagens.

A importância da álgebra de imagens neste trabalho está em seu sentido amplo, pois, sua fundamentação teórica, suas operações e operandos, fornecem a base para as redes neurais morfológicas. Veremos na última seção, como interpretar uma rede neural artificial, em termos desse formalismo algébrico.

O objeto principal desse formalismo teórico, é a definição de imagem, para sua transformação e análise. Uma imagem é definida como sendo uma coleção de pontos e um conjunto de valores associados a esses pontos.

Imagem é portanto uma função  $a$  de  $X$  para  $F$ , sendo  $X$  um conjunto de coordenadas e  $F$  um conjunto de valores. Uma imagem então assume a forma  $a = (x, a(x)) : x \in X$ . Podemos definir cada elemento  $(x, a(x))$  como pixel, em que  $x$  é sua localização e  $a(x)$  seu valor no ponto. Utilizaremos o símbolo  $F^X$  para denotar o conjunto das imagens definidas nas coordenadas  $X$  e que tomam valores no conjunto  $F$ , isto é,  $F^X = \{a : X \rightarrow F\}$  [6]

O conjunto de coordenadas da imagem pode ser qualquer subconjunto  $n$ -dimensional do espaço Euclidiano,  $x = (x_1, x_2, \dots, x_n)$ , enquanto que a escolha para o conjunto de valores é menos restrita e podemos lidar com imagens com conjuntos de valores inteiros, reais, complexos, etc.

O conjunto de valores é uma álgebra homogênea, sendo as operações em e entre seus elementos, as operações usuais associados ao conjunto de valores. Então, se escolhermos como conjunto de valores, os números reais, suas operações binárias serão multiplicação, adição, subtração e divisão, usuais do conjunto real.

Um conjunto de valores importante é a estrutura conhecida como bounded lattice ordered group que assume a forma  $(R, \vee, +, -\infty, 0)$  ou também denotado  $(R_{\pm\infty}, \vee, \wedge, +, +')$ [6]. Esse conjunto de valores é importante, pois, compõe a estrutura algébrica por trás das redes neurais artificiais morfológicas.[9]

Apesar das operações que trataremos em redes neurais artificiais morfológicas serem todas tratadas com valores finitos, para consistência definimos a operação  $+$  em  $R_{\pm\infty}$  da seguinte maneira:

$$a + (-\infty) = (-\infty) + a = -\infty \quad (3)$$

$$a + \infty = \infty + a = \infty \quad (4)$$

$$(-\infty) + \infty = \infty + (-\infty) = -\infty \quad (5)$$

$a \in R_{-\infty}$

A álgebra de imagens é muito apropriada para o estudo de processamento de imagens, pois qualquer algoritmo que pode ser implementado em um computador digital pode também ser expresso em termos de álgebra de imagem.

Além de imagens, também podemos definir outros operandos básicos, como vizinhanças e templates. Templates são os operandos mais interessantes, pois, além de generalizar vizinhanças, máscaras e elementos estruturantes, que estudaremos em morfologia matemática, ainda podem ser interpretados como análogos aos pesos das redes neurais artificiais.

Intuitivamente um template pode ser visto como “uma imagem de imagens”, isto é, para cada ponto de seu domínio  $y$ , o template possui uma imagem associada. De forma mais rigorosa, um template é uma função  $t: Y \rightarrow F^X$ . Portanto,  $t(y)$  é uma imagem com valores de  $F^X$  para todo  $y \in Y$ . De maneira concisa, podemos escrever:

$$t_y = \{(x, t_y(x)) : x \in X\} \quad (6)$$

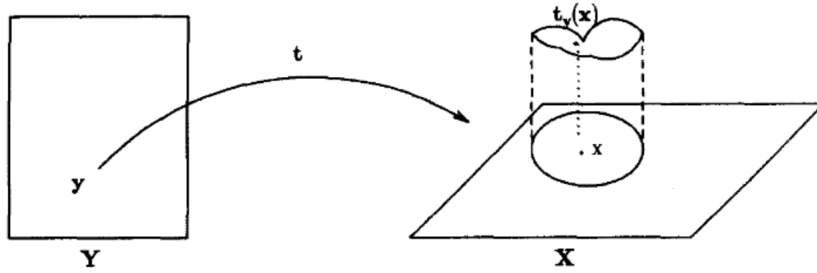


Figura 2: Representação pictórica de um template[12]

Em álgebra de imagens podemos operar imagens com o conjunto de valores  $F$  e operar imagens com imagens, porém, para o estudo de redes neurais artificiais morfológicas, as operações fundamentais serão entre templates e imagens. Estas operações construirão as operações fundamentais em morfologia matemática.

É útil definir o suporte do template  $t_y$ , dessa forma as operações entre imagens e templates ocorrerão na região fora do suporte.

$$S(t_y) = \{x \in X : t_y(x) \neq 0\} \quad (7)$$

Em que 0 representa o elemento neutro da estrutura algébrica utilizada.

Primeiramente podemos definir o produto generalizado entre imagens e templates. Suponha  $X$  finito e  $\gamma$  uma operação binária associativa e comutativa no conjunto de valores  $F$ , então, definiremos a operação  $\Gamma$  em que  $a$  é uma imagem,  $a \in F^X$  como

$$\Gamma_a = \Gamma a(x) = a(x_1)\gamma a(x_2)\gamma \dots \gamma a(x_m) \quad (8)$$

Podemos combinar imagens e templates escolhendo operações binárias apropriadas. Combinando uma operação binária entre templates  $o : F_1 x F_2 \rightarrow F$  e  $\gamma$  operação binária definida acima, podemos induzir a operação generalizada do produto à direita de  $a$  com  $t$ :

$$a \boxtimes t = \{(y, b(y)) : b(y) = \Gamma_{x \in X} a(x) o t_y(x), y \in Y\} \quad (9)$$

Como não especificamos as operações  $\gamma$  e  $o$  fica claro que a escolha dessas, permite definir as mais variadas transformações de imagens. Para o nosso caso, a operação mais importante é o máximo aditivo ( $\boxplus$ ).

Se  $a \in R_{-\infty}^X$  e  $t \in (R_{-\infty}^X)^Y$ , então o máximo aditivo é dado por:

$$a \boxplus t = \{(y, b(y)) : b(y) = \bigvee_{i=1}^N a(x) + t_y(x), y \in Y\} \quad (10)$$

, em que definimos  $\bigvee_{i=1}^N a(x) = t_y(x) = \max\{a(x) + t_y(x) : x \in X\}$ .

De maneira análoga, podemos definir outra importante operação, que é o mínimo aditivo. Seja  $a \in R_{-\infty}^X$  e  $t \in (R_{-\infty}^X)^Y$ , então o mínimo aditivo é dado por:

$$a \boxminus t = \{(y, b(y)) : b(y) = \bigwedge_{i=1}^N a(x) + t_y(x), y \in Y\} \quad (11)$$

, em que definimos  $\bigwedge_{i=1}^N a(x) = t_y(x) = \min\{a(x) + t_y(x) : x \in X\}$ .



## 4 Noções de Morfologia Matemática

A morfologia de imagens surge das operações definidas na teoria de medida geométrica e geometria integral desenvolvidas por Minkowski e Hadwiger:

$$X \ominus B = \bigcap_{b \in B} X_b \quad (12)$$

$$X \oplus \check{B} = \bigcap_{b \in \check{B}} X_b \quad (13)$$

,em que  $\check{B} = \bigcup_{y \in B} \{-y\}$

Serra e Sternberg através dessas operações clássicas formularam os conceitos modernos das transformações morfológicas. No caso, a primeira operação definirá uma erosão e a segunda definirá uma dilatação.

Essas operações também surgem das operações em templates, as operações máximo aditivo e mínimo aditivo, estudadas em templates, são suas generalizações e definem dilatação e erosão, respectivamente.

Uma relação importante entre operações é a dualidade, que é definida em linguagem de álgebra de imagens da seguinte maneira:

$$a \boxtimes t = -(-a \boxminus -t) \quad (14)$$

E como as operações de erosão e dilatação satisfazem tal relação, as chamaremos de operações duais.

Também temos que todas as operações morfológicas são descritas em termos dessas duas operações e portanto, pelo princípio de dualidade, podem ser todas descritas pelo máximo aditivo. Em uma notação parecida com a utilizada por Minkowski e adotada por Serra, temos o dual definido como:

$$X^c \oplus B = (X \ominus B)^c \quad (15)$$

,em que  $X^c$  é o complementar de  $X$ .

Assim temos que dilatar grãos é equivalente a erodir poros e vice-versa. Devemos ser cuidadosos, no entanto, porque apesar de serem operações duais, não são operações inversas. Dilatar e depois erodir uma mesma imagem, não resulta na imagem original.

Observamos também que da tradução da linguagem da álgebra de imagens para a morfologia matemática, o operando template é substituído por um operando chamado elemento de estruturação, no caso acima denotado por  $B$ . Um elemento de estruturação  $B$  também é uma imagem em um espaço métrico qualquer, que é transladada sobre este mesmo espaço métrico com o intuito de caracterizar a nova imagem após a operação morfológica. Por isso, escolher corretamente as características geométricas dos elementos de estruturação é parte significativa da correta aplicação da transformação morfológica.

Dessa maneira, podemos dizer que dilatar  $A$  por  $B$  resulta no conjunto que compreende todos os pontos os quais o elemento de estruturação  $B$  transladado

por  $p$  intersectado por  $A$  não é vazio. E de maneira semelhante dizemos que erodir  $A$  por  $B$  resulta em um conjunto que compreende em todos os pontos os quais o elemento de estruturação  $B$  transladado por  $p$  encaixa inteiramente em  $A$ , em que  $p$  é uma coordenada  $\in X$ .

Através dessas duas operações podemos definir diversas outras operações e transformações morfológicas, a intuição por trás dessas construções parte das propriedades da erosão e da dilatação. Por exemplo, como erodir e depois dilatar não resulta na imagem original, podemos definir as operações de abertura e fechamento. Desse modo, uma abertura de  $A$  por  $B$  é definida como

$$X^c \oplus B = (X \ominus B)^c \quad (16)$$

E um fechamento de  $A$  por  $B$  é definido como:

$$X^c \oplus B = (X \ominus B)^c \quad (17)$$

Outras propriedades dessas operações básicas também inspiraram a criação de outras transformações morfológicas. A propriedade de iteratividade, por exemplo, é usada para definir a dilatação geodésica.

## 5 As Redes Neurais Morfológicas

A principal modificação da rede neural artificial clássica para a introdução da rede neural artificial morfológica é a mudança das operações de multiplicação e adição da equação 1 para operação de adição e máximo (ou mínimo), operações definidas na estrutura  $(R_{\pm\infty}, \vee, \wedge, +, +')$ .

Com isso, podemos diferenciar as duas redes, a clássica da morfológica. Na rede neural clássica, cada neurônio combina as informações, multiplicando os valores de entrada e pesos correspondentes e os somando, enquanto que na rede neural morfológica, os valores de entrada são adicionados aos pesos correspondentes e por fim, faz-se o máximo:

$$y = \sigma\left(\bigvee_{i=1}^N a_i + w_i\right). \quad (18)$$

Nas rede neurais artificiais morfológicas, os pesos do neurônio  $i$  da camada  $j$ , denotado  $w_{ji}$ , pode ser interpretado como um template da álgebra de imagens, dessa forma teríamos  $t_j(i)$ .

É interessante observar que os pesos de valores nulos, são representados pelas regiões fora do suporte do template  $t$  e são ignorados durante as computações. Além disso, podemos representar a função de ativação como:

$$f(a \boxtimes t) = (f_1(a \boxtimes t), f_2(a \boxtimes t), \dots, f_N(a \boxtimes t)). \quad (19)$$

Para  $a \in R_{\pm\infty}^N$  e  $t \in (R_{\pm\infty}^N)^M$ .

Por causa da relação dual que vimos anteriormente, qualquer transformação que seja da forma da equação 10, existe uma rede neural morfológica que realiza os mesmos cálculos.

Assim como no caso das redes neurais artificiais clássicas, a rede neural morfológica também consegue construir uma porta NAND e satisfazer a universalidade da computação, no entanto, ao contrário do caso das redes clássicas, a rede morfológica necessita para tal construção, associações de vários perceptrons. Assim sendo, construindo uma rede neural morfológica de múltiplas camadas, também podemos reproduzir a construção de uma porta XOR, como pode ser visto em Sussner(1998).

Para a construção de uma rede neural morfológica que generalize a dilatação, dado um conjunto de  $N$  entradas  $x = a = (x_1, x_2, \dots, x_N)$ , um conjunto de  $M$  saídas,  $y = (y_1, y_2, \dots, y_M)$  e  $w_{ji} = t_j(i), j = 1, \dots, M, i = 1, \dots, N$  e função de ativação  $\sigma_j(x) = x$ . Em que  $a \in R_{\pm\infty}^N$  e  $t \in (R_{\pm\infty}^N)^M$ . A rede pode ser vista como completamente conectada e para os pesos nulos, simplesmente associamos esses aos valores do template fora do suporte. Dada as outras propriedades construídas ao longo da seção, temos que essa construção é completa e define uma rede generalizada de dilatação.

No entanto, a rede construída acima não possui uma regra de aprendizagem, isto é, para a rede generalizar a operação que gostaríamos, precisaríamos fornecer de antemão os pesos da rede.

Podemos de maneira semelhante construir outras generalizações de operações de morfologia matemática e se fornecidos os pesos, antecipadamente, não precisamos nos preocupar com o aprendizado da rede. Logicamente que os pesos, como vimos, são muito difíceis de serem calculados analiticamente e por isso, precisamos construir estratégias de aprendizado.

No caso usual, que queremos uma rede neural com capacidade de aprendizado, teremos um conjunto de entradas  $a^k$  e um conjunto de saídas esperadas  $d^k$ , definindo um problema de classificação supervisionado. Uma primeira tentativa de implementar o algoritmo na rede neural morfológica com aprendizado pode ser visto em Davidson(1993). Seja o caso de uma rede booleana,  $a \in \{0, 1\}^N$  e  $t \in (\{0, \infty\}^N)^N$  de convergência garantida, com uma função de ativação condizente à dilatação, resultando em um vetor N-dimensional  $c$ , dos valores da saída após as entradas passarem pela rede, em que  $c_j^k = \bigvee_{i \in N(j)} a_i^k + w_{ji}$ ,  $N(j)$  a região de suporte de  $t$ . Então, o sistema de regras de aprendizado definido por Davidson é:

<i>caso</i>	$a_i^k$	$d_j^k$	$c_j^k$	novο valor para $w_{ji}$
1	0	0	1	valor atual
2	0	1	0	valor atual
3	1 ou 0	1	1	valor atual
4	1 ou 0	0	0	valor atual
5	1	0	1	$-\infty$
6	1	1	0	0

Tabela 2: Regras de aprendizagem de pesos para uma rede booleana de dilatação

Por fim, observamos que esse sistema de regras de aprendizagem foi construído intuitivamente e portanto não pode ser estendido para todas as redes neurais artificiais morfológicas, mas somente para as redes booleanas de dilatação construídas por Davidson. Portanto, construir um algoritmo análogo ao algoritmo de Backpropagation para as redes neurais clássicas é fundamental para o progresso da área.

## Referências

- [1] Turing, A. M. Computing Machinery and Intelligence. *Mind* 49: 433-460, 1950.
- [2] Bezdek, J. What is Computational Intelligence?. *Computational Intelligence: Imitating Life*, J.M Zurada, R.J Marks II and C.J Robinson (Eds) IEEE Press. 1-12, 1994.
- [3] Rosenblatt, F. Principles of Neurodynamics. Perceptrons and the Theory of Brain Mechanisms. *Arch Gen Psychiatry*. 7(3):218–219, 1962. doi:10.1001/archpsyc.1962.01720030064010
- [4] McCulloch, W.S.; Pitts, W. *Bulletin of Mathematical Biophysics* 5: 115. <https://doi.org/10.1007/BF02478259>, 1943.
- [5] Goodfellow, I.; Bengio, Y; Courville, A. *Deep Learning*. MIT Press. <http://www.deeplearningbook.org>, 2016.
- [6] Ritter, G.X.; Wilson, J.N. *Handbook of Computer Vision Algorithms in Image Algebra*. 1ª edição. CRC Press. Florida, 1996.
- [7] Nielsen, M. A. *Neural Networks and Deep Learning*. Determination Press, 2015
- [8] Serra, J. *Image Analysis and Mathematical Morphology*. Academic Press. Florida, 1983.
- [9] Ritter, G.X.; Sussner, P. An Introduction to Morphological Neural Networks. *Proceedings of 13th International Conference on Pattern Recognition*, Viena, Áustria. pp. 709-717 vol.4. doi: 10.1109/ICPR.1996.547657, 1996.
- [10] Sussner, P. Morphological Perceptron Learning. *Proceedings of the 1998 IEEE International Symposium on Intelligent Control (ISIC) held jointly with IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA) Intell*, Gaithersburg, MD, USA, pp. 477-482. doi: 10.1109/ISIC.1998.713708, 1998.
- [11] Davidson, J.L.; Hummer, F. Morphology neural networks: An introduction with applications. *Circuits Systems Signal Process*. VOL. 12, N<sup>o</sup>.2, pp. 177-210, 1993.
- [12] Ritter, G.X.; Wilson, J.N.; Davidson, J.L. Image algebra: An overview . *Computer Vision, Graphics, and Image Processing*. Volume 49, Issue 3, pp. 297-331, 1990.
- [13] Blum, A; Rivest, R.L.; Training a 3-node neural network is np-complete. *Neural Networks*. Volume 5, Issue 1, pp. 117-127, 1992.

- [14] Haykin, Simon S.. Neural networks and learning machines. Third Upper Saddle River, NJ: Pearson Education, 2009.
- [15] Gorey, C.; DeepMind team behind AlphaGo wins inaugural ‘Nobel Prize for AI’. Disponível em: <https://www.siliconrepublic.com/machines/deepmind-alphago-marvin-minsky-award>. Acessado em: 20 de Junho de 2019.