

A Comparative Study on Morphological Neural Networks for Binary Classification

Luana Felipe de Barros¹, Marcos Eduardo Valle²

¹Instituto de Computação (IC)
Universidade Estadual de Campinas (Unicamp), Campinas – SP.

²Instituto de Matemática, Estatística e Computação Científica (IMECC)
Universidade Estadual de Campinas (Unicamp), Campinas – SP.

Abstract. *Morphological neural networks represent a class of artificial neural networks whose neurons perform an operation from mathematical morphology followed by the application of an activation function. This paper provides a comparative study of different approaches that use morphological neural networks. Specifically, according to the training rule, we review incremental approaches, approaches based on maximum descent methods, extreme learning machines, and convex-concave optimization procedures. Computational experiments showed that, on average, the reduced dilation-erosion perceptron with bagging and ensemble strategies had better results in several binary classification problems.*

Resumo. *Redes neurais morfológicas representam uma classe de redes neurais artificiais cujos neurônios efetuam uma operação da morfologia matemática seguida da aplicação de uma função de ativação. Este artigo apresenta um estudo comparativo de diferentes abordagens que utilizam redes neurais morfológicas. Especificamente, de acordo com a regra de treinamento, revisamos abordagens incrementais, baseadas no método de máxima descida, máquinas de aprendizado extremo e procedimento de otimização convexa-côncava. Experimentos computacionais mostraram que, em média, o perceptron erosão-dilatação reduzido com as estratégias bagging e ensemble obteve melhores resultados em diversos problemas de classificação binária.*

1. Introdução

Redes neurais artificiais (RNAs, do inglês *artificial neural networks*) são estruturas matemáticas parcialmente inspiradas no cérebro humano, onde as unidades básicas de processamento são os neurônios [Haykin 2009]. Atualmente, as RNAs são amplamente empregadas problemas de reconhecimento de padrões e visão computacional, processamento de linguagem natural, robótica, entre outros. Pode-se dizer que os estudos em RNAs começaram em 1943 quando o biólogo McCulloch e o matemático Pitts introduziram um modelo matemático do neurônio biológico. Aproximadamente 15 anos após a publicação do artigo seminal de McCulloch e Pitts, Rosenblatt propôs uma nova abordagem para o reconhecimento de padrões chamada *perceptron* [Haykin 2009]. O *perceptron* (linear) é composto por uma camada de neurônios de entrada e uma camada de neurônios de saída, totalmente conectadas entre si e ponderados por pesos sinápticos. Cada neurônio de saída corresponde à uma LTU (do inglês *linear threshold unit*): uma unidade que computa uma

soma ponderada dos pesos sinápticos e os valores de entrada. Em seguida, aplica uma função de ativação *hard limiter* à soma. Essas operações são tradicionalmente utilizadas em RNAs clássicas.

Diferente dos modelos tradicionais, Ritter e Sussner introduziram uma nova classe de redes neurais, chamadas redes neurais morfológicas [Ritter and Sussner 1996]. Resumidamente, uma rede neural morfológica realiza uma operação elementar de morfologia matemática em cada neurônio, possivelmente seguida da aplicação de uma função de ativação [Sussner and Esmi 2011]. Em termos algébricos, as operações de multiplicação e adição são substituídas por adição e máximo (ou mínimo), respectivamente, nas redes neurais morfológicas. Essa substituição implica que o cálculo da rede morfológica seja não linear antes da aplicação do limiar. Consequentemente, as propriedades das redes neurais morfológicas são drasticamente diferentes daquelas dos modelos tradicionais de redes neurais [Ritter and Sussner 1996]. É importante destacar que, diferente das redes neurais tradicionais, o treinamento das redes neurais morfológicas é mais difícil devido a não-diferenciabilidade das operações envolvidas. Apresentamos abaixo uma breve revisão das principais contribuições na área de redes neurais morfológicas para classificação de padrões.

Um dos primeiros modelos de redes neurais morfológicas para classificação de padrões foi o perceptron morfológico de camada única (SLMP, do inglês *single-layer morphological perceptron*) [Ritter and Sussner 1996]. Um SLMP realiza ou uma dilatação ou uma erosão da morfologia matemática seguida da aplicação de uma função *hard-limiter*.

O SLMP foi subsequentemente estendido e generalizado por muitos pesquisadores. Por exemplo, Sussner desenvolveu os chamados perceptron morfológico de múltiplas camadas e apresentou um algoritmo supervisionado para classificação binária [Sussner 1998]. Em poucas palavras, o algoritmo proposto por Sussner é um algoritmo incremental no qual são adicionados neurônios na camada oculta até que os dados de treinamento sejam corretamente classificados. Levando em consideração o papel importante dos dendritos nos neurônios biológicos, Ritter e Urcid apresentaram um neurônio morfológico com estrutura de dendritos [Ritter and Urcid 2003]. Apesar da motivação biológica, do ponto de vista matemático, o neurônio morfológico com estrutura de dendritos é semelhante ao perceptron morfológico de múltiplas camadas desenvolvido por Sussner. Com efeito, tal como o algoritmo proposto por Sussner, o neurônio morfológico com estrutura de dendrito cresce gradualmente enquanto houver padrões de treinamento que são erroneamente classificados pela rede neural. Além disso, a função de decisão de ambos, perceptron morfológico de múltiplas camadas e neurônio morfológico com estrutura de dendrito, dependem da ordem no qual os dados são apresentados para a rede neural. O perceptron morfológico com camada competitiva introduzido por Sussner e Esmi, por outro lado, não depende da ordem no qual os dados de treinamento são apresentados [Sussner and Esmi 2011]. Contudo, tal como os modelos morfológicos anteriores, o treinamento do perceptron morfológico com camada competitiva termina apenas quando todos os dados de treinamento são corretamente classificados. Logo, é possível que a rede termine sobre-treinando (*overfitting*) o conjunto de dados.

Além dos algoritmos de treinamento mencionados anteriormente, que podem ser classificados como heurísticas gulosas, o treinamento de redes neurais morfológicas fo-

ram formulados como problemas de otimização. Por exemplo, Pessoa e Maragos usaram funções impulso para evitar a não-diferenciabilidade das operações de máximo e mínimo presentes no neurônio morfológico [Pessoa and Maragos 2000]. Usando as ideias de Pessoa e Maragos, Araujo propôs uma rede híbrida morfológica/linear chamada *perceptron dilatativo-erosivo* (DEP, do inglês *dilation-erosion perceptron*) que foi treinada usando um método de máxima descida [de A. Araújo 2011]. Métodos de máxima descida também foram usados por Hernández et al. para treinar uma rede híbrida com duas camadas; uma morfológica e a outra linear [Hernández et al. 2019]. De um modo similar, Mondal et al. apresentaram uma rede híbrida, chamada rede morfológica densa, que é treinada usando um algoritmo de máxima descida estocástico [Mondal et al. 2019]. Também destacamos que Franchi et al. integraram operadores morfológicos com o paradigma de aprendizado profundo para introduzir as chamadas redes morfológicas profundas, que também são treinadas usando um algoritmo de máxima descida [Franchi et al. 2020].

Diferente dos algoritmos de treinamento baseados no gradiente, Arce et al. treinaram uma rede neural morfológica usando técnicas de computação evolutiva, especificamente evolução diferencial [Arce et al. 2018]. Além disso, Sussner e Campiotti apresentaram uma máquina de aprendizado extremo morfológica/linear composta por uma camada de neurônios morfológicos seguida de uma camada linear que é treinada usando o método dos quadrados mínimos [Sussner and Campiotti 2020].

Finalmente, Charisopoulos and Maragos formularam o treinamento de um SLMP como a solução de um problema de otimização convexa-côncava [Charisopoulos and Maragos 2017]. Além da formulação elegante, o procedimento convexo-côncavo superou métodos de máxima descida em termos de acurácia e robustez em alguns experimentos computacionais. Baseado no trabalho de Charisopoulos e Maragos, Valle combinou máquinas de suporte vetorial com o modelo DEP para problemas de classificação binária [Valle 2020]. Nesse caso, o treinamento do modelo morfológico DEP é efetuado resolvendo um problema de otimização convexo-côncavo.

Concluindo, destacamos anteriormente diferentes abordagens para o treinamento de redes neurais morfológicas com ênfase em problemas de classificação. Apesar da diversidade de arquiteturas e técnicas de treinamento, há ausência de experimentos comparativos do desempenho das redes neurais morfológicas em classificação de padrões. Portanto, a principal motivação para a realização deste trabalho foi completar essa lacuna, apresentando um estudo comparativo entre as principais abordagens que utilizam redes neurais morfológica em problemas reais de classificação binária. O artigo está organizado da seguinte forma: a Seção 2 apresenta fundamentos elementares de redes neurais morfológicas, como também um resumo explicativo das abordagens consideradas. A Seção 3 relata os experimentos computacionais realizados para comparação dos classificadores. O artigo termina com algumas considerações finais na Seção 4.

2. Uma Breve Revisão de Redes Neurais Morfológicas

Nessa seção apresentamos uma breve revisão de diversas redes neurais morfológicas e seus algoritmos de treinamento. Selecionamos modelos representativos de cada arquitetura ou algoritmo de treinamento para classificação binária. Apresentamos exemplos representativos das fronteiras de decisão estabelecidas pelas diferentes redes neurais morfológicas mediante a base de dados MakeMoons.

Em um problema de classificação binária, uma rede neural atua como uma função de características $\mathcal{X} \subseteq \mathbb{R}^n$ no conjunto das classes $\mathcal{C} = \{c_0, c_1\}$. O conjunto das classes é geralmente identificado como $\mathcal{C} = \{0, 1\}$ ou $\mathcal{C} = \{-1, +1\}$. Finalmente, os pesos sinápticos e outros hiper-parâmetros são determinados utilizando um conjunto de treinamento definido por:

$$\Gamma = \{(\mathbf{x}_i, d_i) : i = 1, \dots, k\} \subseteq \mathcal{X} \times \mathcal{C}, \quad (1)$$

em que $\mathbf{x}_i \in \mathcal{X}$ é o vetor de características e $d_i \in \mathcal{C}$ é a classe da i -ésima amostra de treinamento, para $i = 1, \dots, k$.

2.1. Perceptron Morfológico de Camada Única (SLMP)

O classificador SLMP foi proposto por Ritter e Sussner como uma alternativa para o perceptron linear de Rosenblatt [Ritter and Sussner 1996]. Os neurônios de uma SLMP realizam ou uma dilatação ou uma erosão da morfologia matemática seguida da aplicação de uma função *hard-limiter*.

A dilatação é obtida calculando o máximo da soma entre os valores de entrada e pesos sinápticos. A erosão, por sua vez, é dada pelo mínimo desta mesma soma. Em termos matemáticos, os operadores de erosão e dilatação aplicados num vetor de característica $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ são definidos pelas equações

$$\varepsilon_w(\mathbf{x}) = \min_{j=1:n} (w_j + x_j) \quad \text{e} \quad \delta_m(\mathbf{x}) = \max_{j=1:n} (m_j + x_j), \quad (2)$$

respectivamente, em que w_1, \dots, w_n e m_1, \dots, m_n representam os pesos sinápticos. Dessa forma, um SMLP é caracterizado por uma das seguintes equações:

$$y = f(\varepsilon_w(\mathbf{x})) \quad \text{ou} \quad y = f(\delta_m(\mathbf{x})), \quad \forall \mathbf{x} \in \mathbb{R}^n, \quad (3)$$

em que f denota uma função de ativação *hard-limiter* tal como a função degrau por:

$$f(z) = \begin{cases} 1, & z > 0, \\ 0, & \text{caso contrário.} \end{cases} \quad (4)$$

O processo de treinamento da SLMP consiste em encontrar as conexões sinápticas w_j ou m_j , de modo que cada amostra seja classificada como 0 ou 1. Dado um conjunto de treinamento Γ dado por (1) com $\mathcal{C} = \{0, 1\}$, os pesos sinápticos de um neurônio de dilatação são determinados de forma direta através da equação

$$m_j = \min\{-x_{ij} : d_i = 0\} = -\max\{x_{ij} : d_i = 0\}, \quad \forall j = 1, \dots, n. \quad (5)$$

Similar ao neurônio de dilatação, os pesos sinápticos de um neurônio de erosão são determinados através da equação

$$w_j = \max\{-x_{ij} : d_i = 0\} = -\min\{x_{ij} : d_i = 0\}, \quad \forall j = 1, \dots, n. \quad (6)$$

Após a obtenção dos pesos sinápticos, é possível classificar um vetor de característica \mathbf{x} entre as classes 0 ou 1 usando (3). Geometricamente, a região de decisão de um

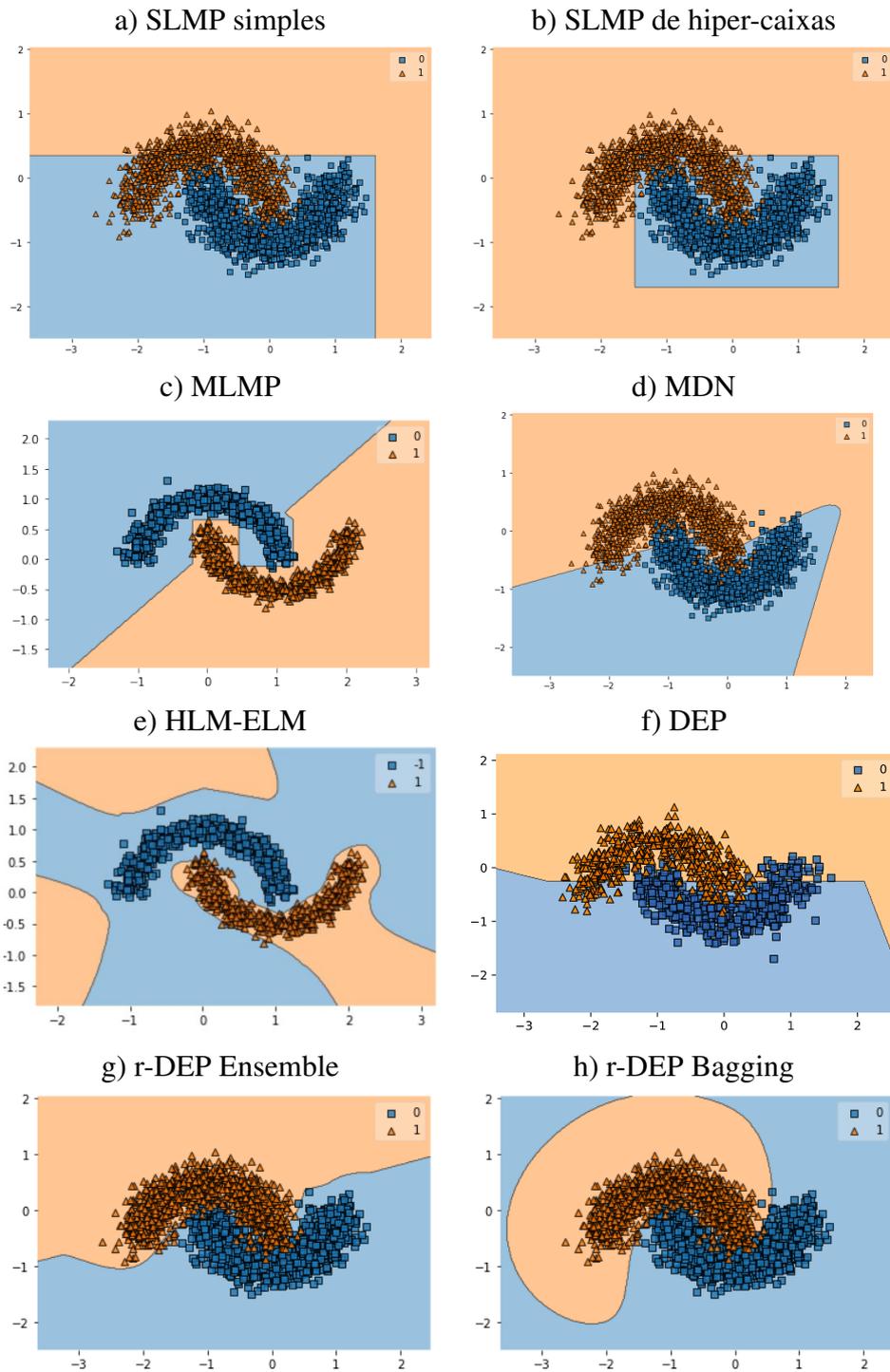


Figura 1. Fronteiras de decisão no conjunto de dados sintético *MakeMoons*.

SMLP é $\{\mathbf{x} \in \mathbb{R}^n : x_i \leq -w_i, \forall i = 1, \dots, n\}$ ou $\{\mathbf{x} \in \mathbb{R}^n : x_i \geq -m_i, \forall i = 1, \dots, n\}$. A Figura 1a) ilustra a fronteira de decisão de uma rede SMLP determinada usando o conjunto de dados sintéticos *MakeMoons*.

Note que, dependendo da distribuição dos dados no espaço das características \mathcal{X} , a fronteira de decisão pode não ser suficiente para separar as amostras de cada classe. A

SLMP de hiper-caixa apresenta uma alternativa muito simples para superar algumas das limitações da SLMP simples. Especificamente, diferente da SLMP simples, a fronteira de decisão de uma SLMP de hiper-caixa é uma hiper-caixa que contém todos os pontos de uma certa classe. Por exemplo, o interior da hiper-caixa pode corresponder à classe 0 enquanto o exterior corresponde à classe 1.

Em termos matemáticos, sejam $-\mathbf{u}$ e \mathbf{v} os vértices inferior e superior da hiper-caixa. O interior da hiper-caixa que corresponde à classe 0 é dada pelo conjunto de todos os vetores $\mathbf{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$ tais que

$$-u_j \leq x_j \leq v_j, \quad \forall j = 1, \dots, n. \quad (7)$$

Usando um conjunto de treinamento Γ , os vértices inferior e superior que contém apenas dados da classe 0 podem ser facilmente determinados usando as equações

$$-\mathbf{u} = \min\{\mathbf{x}_i : d_i = 0\} \quad \text{e} \quad \mathbf{v} = \max\{\mathbf{x}_i : d_i = 0\}. \quad (8)$$

Além disso, a superfície de decisão do SLMP de hiper-caixa pode ser definida utilizando a operação de erosão da morfologia matemática. Especificamente, a classificação é dada por $\psi(\mathbf{x}) \geq 0$ em que

$$\psi(\mathbf{x}) = \min\{[\mathbf{u}, \mathbf{v}] + [\mathbf{x}, -\mathbf{x}]\}, \quad \forall \mathbf{x} \in \mathbb{R}^n. \quad (9)$$

Com efeito, $\psi(\mathbf{x}) \geq 0$ se, e somente se, $\mathbf{u} + \mathbf{x} \geq 0$ e $\mathbf{v} - \mathbf{x} \geq 0$, ou ainda, $-u_j \leq x_j$ e $x_j \leq v_j$ para todo $j = 1, \dots, n$. Portanto, $\psi(\mathbf{x}) \geq 0$ se, e somente se, \mathbf{x} satisfaz (7). Um modelo baseado na dilatação pode ser obtido de modo similar.

A Figura 1b) mostra a fronteira de decisão do modelo SLMP de hiper-caixa dado por (9). Os pesos sinápticos \mathbf{u} e \mathbf{v} foram determinados por (8) usando o conjunto de dados sintéticos *MakeMoons*.

2.2. Perceptron Morfológico com Camada Competitiva (MLMP)

A SLMP de hiper-caixa possui uma limitação geométrica – os dados de uma classe devem estar contidos numa hiper-caixa. Para contornar este problema, um modelo com múltiplas hiper-caixas (MLMP) que delimitam melhor as amostras de treinamento foi proposto por Sussner e Esmi [Sussner and Esmi 2011]. A ideia é utilizar um processo recursivo da seguinte forma para cada classe:

1. Determine a menor hiper-caixa H_0 que contém todas as amostras da classe 0.
2. Determine também a menor hiper-caixa H_1 que contém todas as amostras da classe 1.
3. Calcule a hiper-caixa obtida pela intersecção $H = H_1 \cap H_2$.
4. Determine hiper-caixas que classificam corretamente as amostras que estão fora da caixa H .
5. Repete-se o processo para as amostras no interior de H .

Finalmente, uma amostra de teste é classificada como sendo da classe da hiper-caixa mais próxima. A fronteira de decisão do classificador MLMP para o conjunto *MakeMoons* é mostrado na Figura 1c).

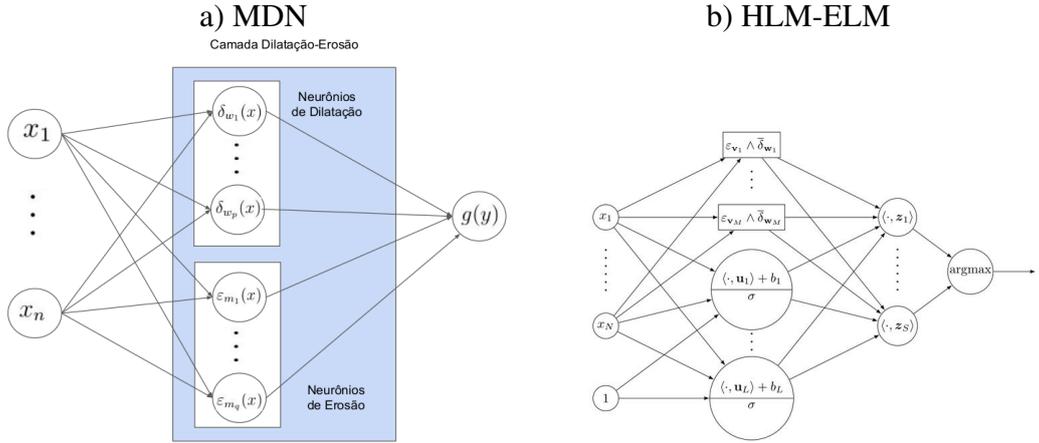


Figura 2. Arquitetura das redes híbridas morfológicas/linear MDN e HLM-ELM.

2.3. Rede Morfológica Densa (MDN)

Recentemente, Mondal et al. apresentaram uma rede neural que utiliza blocos de neurônios morfológicos em uma camada de dilatação-erosão, chamada rede morfológica densa (MDN, do inglês *morphological dense network*) [Mondal et al. 2019]. A MDN possui uma arquitetura com uma camada de dilatação-erosão e uma camada linear de saída. A camada de dilatação-erosão possui p neurônios de dilatação e q neurônios de erosão dados por (2), tal como ilustrado na Figura 2a). Ainda, temos y como sendo a saída da camada morfológica. Num problema de classificação binária, considera-se apenas um neurônio linear com uma função de ativação sigmoide na camada de saída.

Como em outras redes neurais, cada neurônio de uma camada tem pesos sinápticos associados a neurônios da camada anterior, que são atualizados no processo de treinamento pelo algoritmo *backpropagation*. A função perda Entropia Cruzada Binária pode ser utilizada em problemas de classificação binária. O resultado para o teste com $p = 128$ neurônios de dilatação e $q = 128$ neurônios de erosão pode ser visto na Figura 1d).

2.4. Máquina de Aprendizado Extremo Híbrida Morfológico/Linear (HLM-ELM)

Vamos agora apresentar a máquina de aprendizado extremo híbrida morfológico/linear (HLM-ELM, em inglês *hybrid linear/morphological extreme learning machine*) proposta recentemente por Sussner e Campiotti [Sussner and Campiotti 2020]. De um modo geral, uma máquina de aprendizado extremo é uma técnica de treinamento generalizado para redes neurais *feedforward* com neurônios lineares na camada de saída e, geralmente, com apenas uma única camada oculta [Huang et al. 2015]. Os pesos sinápticos e bias dos neurônios da(s) camada(s) oculta(s) são inicializados de forma aleatória. Já os parâmetros da camada de saída são determinados utilizando o método dos quadrados mínimos. Os benefícios de utilizar a máquina de aprendizado extremo incluem a capacidade de aproximação universal e ganho de eficiência, visto que os pesos da camada oculta não precisam ser aprendidos, como mostrado em [Huang et al. 2015].

A rede HLM-ELM possui uma camada oculta com p neurônios morfológicos de hiper-caixas, denotados por ψ_1, \dots, ψ_p e descrito por (9). Em paralelo, a camada oculta também possui q neurônios lineares com função de ativação sigmoide dados por

$$g_j(\mathbf{x}) = \sigma(\langle \mathbf{w}_j, \mathbf{x} \rangle + b_j), \quad \forall j = 1, \dots, q, \quad (10)$$

com $\sigma(t) = 1/(1 + e^{-t})$. Dessa forma, a saída da camada oculta define uma função $\mathbf{h} : \mathbb{R}^n \rightarrow \mathbb{R}^{p+q+1}$ dada por:

$$\mathbf{h}(\mathbf{x}) = [1, \psi_1(\mathbf{x}), \dots, \psi_M(\mathbf{x}), g_1(\mathbf{x}), \dots, g_L(\mathbf{x})]. \quad (11)$$

Num problema de classificação binária, a camada de saída é composta por um único neurônio linear. Especificamente, a função de decisão $\tau : \mathbb{R}^n \rightarrow \mathbb{R}$ da HLM-ELM é definida pelo produto interno dos vetores $\boldsymbol{\alpha} \in \mathbb{R}^{p+q+1}$ e $\mathbf{h}(\mathbf{x})$, isto é,

$$\tau(\mathbf{x}) = \langle \boldsymbol{\alpha}, \mathbf{h}(\mathbf{x}) \rangle. \quad (12)$$

A arquitetura do modelo HLM-ELM pode ser visualizada na Figura 2b).

O treinamento de uma HLM-ELM é efetuado como segue:

1. Primeiro, os pesos sinápticos dos neurônios da camada oculta são gerados aleatoriamente. Por um lado, os pesos dos neurônios morfológicos de hiper-caixa são definidos aleatoriamente dentro de um intervalo $[a, b]$, onde a e b representam as amostras mínimas e máximas do conjunto de dados Γ . Por outro lado, os neurônios lineares têm pesos definidos aleatoriamente em uma distribuição normal.
2. Posteriormente, determina-se o vetor $\boldsymbol{\alpha} \in \mathbb{R}^{p+q+1}$ resolvendo o problema de quadrados mínimos

$$\min \sum_{i=1}^k \|\tau(\mathbf{x}_i) - d_i\|^2 \quad (13)$$

com $d_i \in \{-1, +1\}$.

Na Figura 1e) apresentamos a visualização da fronteira de decisão obtida por uma HLM-ELM com 10 neurônios morfológicos e 15 neurônios lineares no conjunto de dados sintético *MakeMoons*.

2.5. Perceptron Erosão-Dilatação (DEP) com Otimização Convexo-Côncavo

O *Perceptron Erosão-Dilatação* (DEP, do inglês *dilation-erosion perceptron*) é obtido combinando neurônios morfológicos de erosão e dilatação com uma soma convexa [de A. Araújo 2011]. Em termos matemáticos, a função de decisão de um modelo DEP aplicado para problemas de classificação binária é descrita pela equação

$$\tau(\mathbf{x}) = \beta \varepsilon_w(\mathbf{x}) + (1 - \beta) \delta_m(\mathbf{x}), \quad \forall \mathbf{x} \in \mathbb{R}^n \quad (14)$$

em que $0 \leq \beta \leq 1$ é um parâmetro que controla a *trade-off* entre os neurônios de erosão e dilatação, e w . A decisão é feita considerando a função *hard-limiter* descrita por:

$$f(z) = \begin{cases} +1, & z \geq 0, \\ -1, & \text{caso contrário.} \end{cases} \quad (15)$$

Desta forma, cada amostra do conjunto de dados é mapeada à classe +1 ou -1. O classificador binário DEP é dado por $y = f(\tau(\mathbf{x}))$.

Podemos inferir que um neurônio de erosão foca em classificar as amostras positivas, enquanto que um neurônio de dilatação foca em classificar amostras negativas.

Portanto, uma combinação dos dois neurônios pode gerar melhores fronteiras de decisão. Para determinar os pesos sinápticos e o parâmetro β , primeiramente dividimos as amostras positivas e negativas nos conjuntos P e N , respectivamente:

$$P = \{\mathbf{x}_i : (\mathbf{x}_i, d_i) \in \Gamma, d_i = +1\} \quad \text{e} \quad N = \{\mathbf{x}_i : (\mathbf{x}_i, d_i) \in \Gamma, d_i = -1\} \quad (16)$$

Ainda, considere ψ_u a função de decisão tanto da erosão quanto da dilatação, ou seja, $\psi_u = \varepsilon_w$ com $\mathbf{u} = \mathbf{w}$ ou $\psi_u = \delta_m$ com $\mathbf{u} = \mathbf{m}$. Um método de otimização convexo-côncavo é usado para determinar \mathbf{u} , que é definido como a solução do seguinte problema de otimização [Charisopoulos and Maragos 2017, Valle 2020]:

$$\min_{\mathbf{u}, \boldsymbol{\xi}} J(\mathbf{u}, \boldsymbol{\xi}) = \frac{1}{|N|} \sum_1^{|N|} v_i^- \max\{0, \xi_i^-\} + \frac{1}{|P|} \sum_1^{|P|} v_i^+ \max\{0, \xi_i^+\} + C \|\mathbf{u} - \mathbf{r}\| \quad (17)$$

$$\text{sujeito a } \psi_u(\mathbf{x}_i) \leq \xi_i^-, \quad \forall \mathbf{x}_i \in N, \quad \text{e} \quad \psi_u(\mathbf{x}_i) \leq -\xi_i^+, \quad \forall \mathbf{x}_i \in P, \quad (18)$$

em que C é um parâmetro de regularização, \mathbf{r} é um valor de referência para \mathbf{u} , ξ_i^- e ξ_i^+ são variáveis de folga no problema de otimização e v_i^- e v_i^+ são seus pesos de penalidade que são utilizados para penalizar valores discrepantes. As variáveis de folga são utilizadas como majorantes do erro de classificação. O objetivo é utilizá-las para encontrar pesos sinápticos nos quais a função de decisão das classes negativas sejam menores que 0 e das positivas maiores ou iguais a 0.

Dito isto, o treinamento do classificador DEP utiliza uma heurística gulosa, isto é, o problema de otimização para encontrar os pesos sinápticos \mathbf{w} e \mathbf{m} do neurônio de erosão e dilatação de forma independente. Em seguida, procura a melhor combinação convexa para o problema determinando o parâmetro β otimizando a função de perda:

$$\min_{0 \leq \beta \leq 1} H(\beta) = \sum_{i=1}^k \max\{0, -d_i \tau(\mathbf{x}_i)\} \quad (19)$$

A Figura 1f) mostra a fronteira de decisão encontrada com o classificador DEP.

2.6. Perceptron Erosão-Dilatação Reduzido (r-DEP)

Visto que as operações de erosão e dilatação são crescentes, o classificador DEP preserva a ordem das classes. Em outras palavras, se uma amostra \mathbf{x} pertence a classe negativa e uma amostra \mathbf{y} pertence a classe positiva, então o vetor de característica de \mathbf{x} deve ser menor que o vetor de característica de \mathbf{y} . Matematicamente, $\mathbf{x} \leq \mathbf{y}$ implica $\psi(\mathbf{x}) \leq \psi(\mathbf{y})$. Portanto, há uma hipótese sobre as características do conjunto de dados que, quando insatisfeita, leva a erros de classificação. Por conta disso, Valle introduziu uma abordagem baseada na morfologia matemática multivalorada e máquinas de suporte vetorial (SVM, do inglês *support vector machines*) chamada perceptron erosão-dilatação reduzido (r-DEP, do inglês *reduced dilation-erosion perceptron*) [Valle 2020].

De um modo geral, SVMs são usadas para definir transformações nos dados de modo a garantir que estes sejam ordenáveis em um novo espaço de características. Especificamente, um classificador SVM (SVC, do inglês *support vector classifier*) pode ser usado para classificação binária e visa maximizar a margem de separação dos dados

através de um hiperplano num espaço transformado. A ideia principal do modelo r-DEP é utilizar a função de decisão de diferentes SVCs para depois aplicar o classificador DEP. Com efeito, a função de decisão de um SVC está relacionada à distância da amostra de entrada ao hiperplano definido no processo de treinamento. Se a distância for positiva, significa que a amostra está acima da curva/hiperplano e, se for negativa, abaixo. Desta forma, a função de decisão do classificador SVC garante que os dados fiquem ordenados.

Sobretudo, a função de decisão de r SVCs define uma aplicação $\rho : \mathbb{R}^n \rightarrow \mathbb{R}^r$. Em termos matemáticos, a função de decisão τ^r de um r-DEP é dada pela composição $\tau^r = \tau \circ \rho$, em que τ é a dada por (14). Finalmente, dada a função f é a *hard-limiter* dada por (15), o r-DEP classifica uma amostra \mathbf{x} usando a equação $y = f(\tau^r(\mathbf{x}))$.

A aplicação ρ de um r-DEP pode ser determinada utilizando as abordagens *ensemble* ou *bagging*, conforme descrito abaixo:

- **Ensemble**: Nesta abordagem os classificadores SVC são determinados utilizando o mesmo conjunto de treinamento dado por (1), porém com diferentes *kernels*.
- **Bagging**: Nesta abordagem os classificadores SVC são determinados utilizando o mesmo *kernel*, porém com conjuntos distintos de amostras do conjunto de treinamento.

Os *kernels* utilizados no *ensemble* foram: linear, função de base radial (RBF, do inglês *radial basis function*) e polinomial. No *bagging*, consideramos 10 SVCs distintos mas com o kernel RBF.

A Figura 1g) mostra a fronteira de decisão encontrada com r-DEP utilizando a técnica *ensemble* com os kernels RBF e linear. Analogamente, a Figura 1h) mostra a fronteira de decisão encontrada com r-DEP utilizando a técnica *bagging* com kernel RBF.

3. Experimentos Computacionais

Para avaliar o desempenho geral dos modelos baseados em redes neurais morfológicas, consideramos 30 problemas de classificação binária disponíveis no repositório OpenML (disponível em <https://www.openml.org/>). Além disso, lidamos com dados ausentes usando o comando `SimpleImputer` da biblioteca `sklearn` em `python`. A lista de bases de dados foi a mesma utilizada em [Valle 2020]. Para avaliar quantitativamente os métodos calculamos a acurácia balanceada combinada com a técnica de validação cruzada estratificada com 5 pastas (*5-folds*). Em seguida, calculamos a média da acurácia balanceada das 5 pastas para cada classificador em cada base de dados. Além dos classificadores baseados em redes neurais morfológicas descritos na seção anterior, incluímos em nossa comparação a técnica de *bagging* para o modelo MLMP, visto que o mesmo é muito sensível a valores discrepantes. No entanto, essa técnica não demonstrou melhora significativa no desempenho do modelo. Ainda, incluímos os classificadores SVC com kernels linear e RBF, respectivamente.

A Figura 3 apresenta o diagrama de caixa obtido com as acurácias balanceadas dos classificadores nas 30 bases de dados consideradas. Com isso, podemos inferir que os classificadores SLMP e SLMP de hiper-caixas não conseguem reproduzir bem a distribuição dos dados para classificação. Ainda, é possível observar que para essas bases de dados de classificação binária, os modelos RBF-SVC e r-DEP quando aplicados a técnica de *ensemble* (Linear SVC e RBF SVC) e *bagging* (RBF SVC) tiveram desempenho médio superiores aos demais, cerca de 85% de acertos. Confirmamos o desempenho

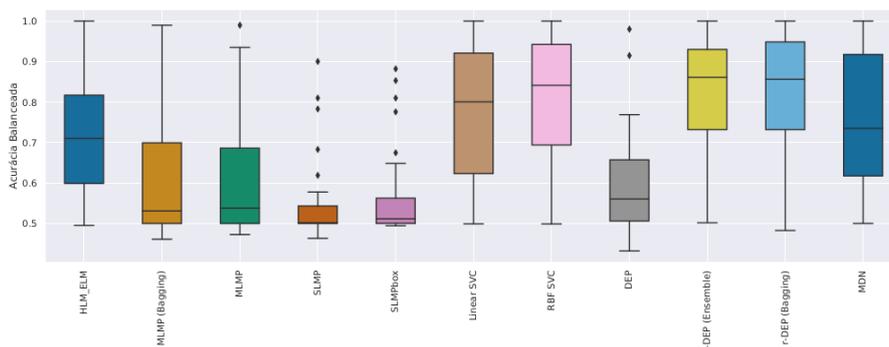
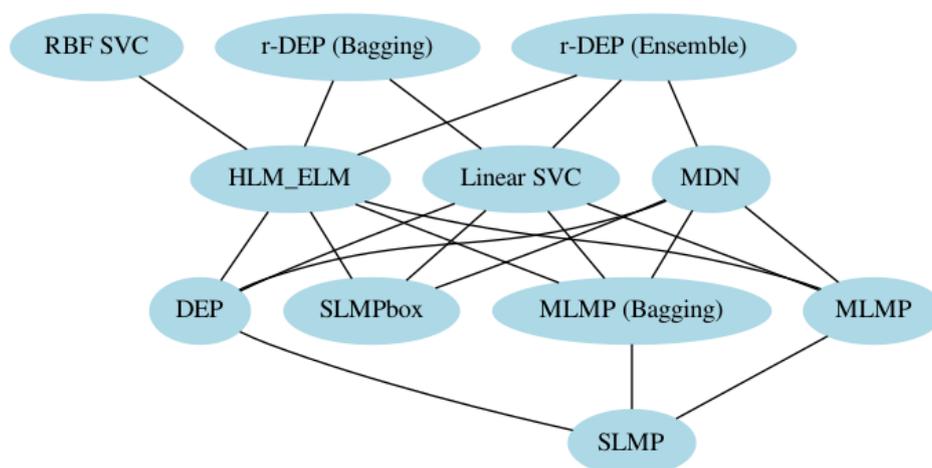


Figura 3. Diagrama de caixa da acurácia balanceada pelos classificadores binários.



Hasse diagram of Wilcoxon signed-rank test (confidence level at 99.0%)

Figura 4. Diagrama de Hasse comparando os classificadores.

superior dos modelos RBF-SVC e r-DEP usando testes estatísticos. Especificamente, a Figura 4 apresenta o diagrama de Hasse construído aplicando o teste de Wilcoxon com nível de confiança 99%. Aqui, SLMPbox corresponde ao SLMP de hiper-caixas. Nessa figura, uma aresta informa que o método no vértice acima superou estatisticamente o modelo no vértice abaixo. Note que os modelos SVC RBF bem como o r-DEP bagging e ensemble superam os demais classificadores nos experimentos realizados.

4. Conclusão

Este artigo apresentou uma breve revisão de diversos modelos de redes neurais morfológicas. Sobretudo, foi apresentado um estudo comparando as diferentes abordagens em 30 problemas de classificação binária. O experimento computacional mostrou que os classificadores RBF SVC e r-DEP com as estratégias *bagging* e *ensemble* tiveram resultados competitivos e melhores que os demais modelos. O repositório com os experimentos realizados está disponível em <https://github.com/httpplups/morphological-networks>.

Referências

- [Arce et al. 2018] Arce, F., Zamora, E., Sossa, H., and Barrón, R. (2018). Differential evolution training algorithm for dendrite morphological neural networks. *Applied Soft Computing*, 68:303–313.
- [Charisopoulos and Maragos 2017] Charisopoulos, V. and Maragos, P. (2017). Morphological perceptrons: Geometry and training algorithms. In Angulo, J., Velasco-Forero, S., and Meyer, F., editors, *Mathematical Morphology and Its Applications to Signal and Image Processing*, pages 3–15, Cham. Springer International Publishing.
- [de A. Araújo 2011] de A. Araújo, R. (2011). A class of hybrid morphological perceptrons with application in time series forecasting. *Knowledge-Based Systems*, 24(4):513 – 529.
- [Franchi et al. 2020] Franchi, G., Fehri, A., and Yao, A. (2020). Deep morphological networks. *Pattern Recognition*, 102:107246.
- [Haykin 2009] Haykin, S. (2009). *Neural Networks and Learning Machines*. Prentice-Hall, Upper Saddle River, NJ, 3rd edition edition.
- [Hernández et al. 2019] Hernández, G., Zamora, E., Sossa, H., Téllez, G., and Furlán, F. (2019). Hybrid neural networks for big data classification. *Neurocomputing*, page S0925231219314560.
- [Huang et al. 2015] Huang, G., Huang, G. B., Song, S., and You, K. (2015). Trends in extreme learning machines: A review.
- [Mondal et al. 2019] Mondal, R., Santra, S., and Chanda, B. (2019). Dense morphological network: An universal function approximator.
- [Pessoa and Maragos 2000] Pessoa, L. and Maragos, P. (2000). Neural networks with hybrid morphological/rank/linear nodes: a unifying framework with applications to handwritten character recognition. *Pattern Recognition*, 33:945–960.
- [Ritter and Sussner 1996] Ritter, G. X. and Sussner, P. (1996). An Introduction to Morphological Neural Networks. In *Proceedings of the 13th International Conference on Pattern Recognition*, pages 709–717, Vienna, Austria.
- [Ritter and Urcid 2003] Ritter, G. X. and Urcid, G. (2003). Lattice Algebra Approach to Single-Neuron Computation. *IEEE Transactions on Neural Networks*, 14(2):282–295.
- [Sussner 1998] Sussner, P. (1998). Morphological perceptron learning. In *IEEE International Symposium on Intelligent Control - Proceedings*, pages 477–482. IEEE.
- [Sussner and Campiotti 2020] Sussner, P. and Campiotti, I. (2020). Extreme learning machine for a new hybrid morphological/linear perceptron. *Neural Networks*, 123:288 – 298.
- [Sussner and Esmi 2011] Sussner, P. and Esmi, E. L. (2011). Morphological Perceptrons with Competitive Learning: Lattice-Theoretical Framework and Constructive Learning Algorithm. *Information Sciences*, 181(10):1929–1950.
- [Valle 2020] Valle, M. E. (2020). Reduced dilation-erosion perceptron for binary classification. *Mathematics*, 8(4):512.