Uma formulação para a máquina de aprendizagem mínima baseada em programação linear

Tamara Arruda Pereira[1], **Amauri Holanda de Souza Júnior**[1]

[1]Departamento de Ciência da Computação
Instituto Federal do Ceará (IFCE)
Maracanaú – Ceará – Brasil

tamaraarrudap@gmail.com, amauriholanda@ifce.edu.br

Abstract. *Minimal Learning Machine (MLM) is a supervised learning method whose basic principle is based on a linear mapping between distances in the input and output spaces, followed by an optimization process to, based on estimated distances, provide an estimate for the output in a typical regression case. The MLM test step involves solving a non-convex optimization problem and it may suffer from local minima problems. In this paper, we present a formulation for the out-of-sample step using linear programming.The experiments show that the proposed method achieves similar performance to that obtained with the original algorithm, additionally producing results with small variance.*

Resumo. *A máquina de aprendizagem mínima (MLM) é um método de aprendizado supervisionado que consiste na utilização de um mapeamento linear entre distâncias dos espaços de entrada e saída, seguido de um processo de otimização para, a partir das distâncias estimadas, estimar a saída. A etapa de teste da MLM envolve a resolução de um problema de otimização não-convexo, e pode sofrer com problemas associados a mínimos locais. Com isso em vista, neste artigo é apresentada uma formulação nessa etapa utilizando programação linear. Os experimentos mostram que o método proposto atinge desempenho semelhante àquele obtido com o algoritmo original, adicionalmente produzindo resultados com menor variância.*

1. Introdução

Na sua essência, aprendizagem de máquina (*machine learning*) é uma subárea da inteligência artificial que compreende técnicas estatísticas ou bioinspiradas na construção de modelos matemáticos capazes de lidar com incertezas a partir de dados [Hastie et al. 2001, Bishop et al. 2006]. Dentre aplicações clássicas, cita-se reconhecimento de voz [Sainath et al. 2015] e imagens [Krizhevsky et al. 2012], jogos [Silver et al. 2016] e robótica [Lenz et al. 2015].

Dentre os principais paradigmas utilizados em aprendizagem de máquina estão métodos supervisionados e métodos não-supervisionados. Na abordagem supervisionada, há variáveis de entrada e saída, o objetivo é encontrar o mapeamento, em geral não-linear, que relaciona essas variáveis. No paradigma não-supervisionado, um dos principais objetivos é encontrar representações úteis a partir somente dos

dados disponíveis pelas variáveis de entrada. Tarefas típicas em aprendizagem não-supervisionada são *manifold learning*, análise de agrupamentos, e aprendizagem de representação (*representation learning*) [Hinton and Salakhutdinov 2006]. Por outro lado, regressão e classificação são exemplos de tarefas de aprendizagem supervisionada. Em regressão, as variáveis de saída são dadas por números reais (contínuas) ao passo que em problemas de classificação as saídas são dadas por categorias ou classes (conjunto discreto finito).

Recentemente, um novo método chamado máquina de aprendizagem mínima (*Minimal Learning Machine*, MLM) [Souza Júnior et al. 2015] tem ganhado atenção. Máquina de aprendizagem mínima é um método supervisionado baseado em mapeamentos entre distâncias computadas no espaço de entrada e saída. Ele pode ser dividido em duas etapas: mapeamento entre distâncias (treinamento) e estimação da saída (teste). Na primeira etapa, utiliza-se um modelo linear multiresposta entre distâncias computadas para pontos fixos chamados pontos de referência entre variáveis de entrada e saída. O modelo linear é estimado através do método dos mínimos quadrados e essa etapa consiste no treinamento do modelo. A etapa de teste consiste em encontrar estimativas para a variável de saída associada a pontos ainda não utilizados na etapa de treinamento do modelo. Para isso, a MLM originalmente emprega um método de otimização que a partir da estimativas de distâncias no espaço de saída encontra a verdadeira localização dos pontos (variáveis), em um procedimento conhecido como *multilateration*. Dentre as vantagens da MLM estão a necessidade de ajustar um único hyperparamêtro (número de pontos de referência); a facilidade de implementação; e o baixo custo computacional para treinamento do modelo. No entanto, a etapa de teste possui um alto custo computacional visto que um problema de otimização não-convexo deve ser resolvido.

Este trabalho propõe a utilização de um novo método para solucionar o problema de otimização que faz parte da etapa de teste da MLM. Mais especificamente, o problema de otimização, originalmente não-convexo, conhecido como *multilateration*, é transformado em um problema de programação linear com solução única e global. Experimentos são realizados para mostrar o potencial da técnica proposta em problemas de regressão clássicos na área de aprendizado de máquina.

O restante do artigo está organizado da seguinte forma: na seção 2 são descritos os fundamentos da MLM conforme a proposta original; a seção 3 apresenta o método proposto neste trabalho: máquina de aprendizagem mínima com programação linear; a seção 4 apresenta e discute os resultados alcançados; e as conclusões são dadas na seção 5.

2. Máquina de Aprendizagem Mínima

Defina o problema de aprendizagem de máquina como o problema de aproximar uma função alvo suave contínua $f : \mathcal{X} \to \mathcal{Y}$ a partir de dados $\mathcal{D} = \{(x_n, y_n = f(x_n))\}_{n=1}^{N}$, onde $x_n \in \mathcal{X}$ e $y_n \in \mathcal{Y}$. Assumindo $\mathcal{X}$ e $\mathcal{Y}$ como sendo o espaço de entrada e saída, respectivamente.

A MLM tem como objetivo aproximar a função alvo f através do uso de funções auxiliares $\delta_k : \mathcal{Y} \to \mathbb{R}_+$ e $\phi_k : \mathcal{X} \to \mathbb{R}_+$ no espaço de saída e entrada, respectivamente. As funções auxiliares são distâncias computadas a partir de pontos

fixos $\{(m_k, t_k = f(m_k)) \in \mathcal{X} \times \mathcal{Y}\}_{k=1}^{K}$, também chamados de pontos de referência. Para simplificar a notação, usa-se $\phi_k(x) = \phi(x, m_k)$ e $\delta_k(y) = \delta(y, t_k)$. A partir de agora, assume-se $\mathcal{X} = \mathbb{R}^D$ e $\mathcal{Y} = \mathbb{R}^S$ e denota-se o conjunto $\{m_k\}_{k=1}^{K}$ como pontos de referência de entrada, e $\{t_k\}_{k=1}^{K}$ como os correspondentes pontos de referência de saída.

Considere a existência de um mapeamento g_k entre os espaços induzidos pelas funções de distância δ e ϕ tal que $g_k : \prod_{j=1}^{K} \phi_j(\mathcal{X}) \to \delta_k(\mathcal{Y})$. As distâncias ponto-a-ponto calculadas entre os dados e os pontos de referência no espaço de entrada são armazenados na matriz $\Phi \in \mathbb{R}^{N \times K}$. De forma similar, considere a matriz de distância na saída entre as N amostras de treinamento e os pontos de referência dada por $\Delta \in \mathbb{R}^{N \times K}$. Usando os dados, pode-se expressar o mapeamento g_k através do modelo $\Delta_{n,k} = g_k(\Phi_{n,\cdot}) + \epsilon_n$ para todo $n = 1, \ldots, N$. O termo $\Phi_{n,\cdot}$ denota todas as colunas da n-ésima linha da matriz Φ e ϵ_n representa o resíduo no contexto de regressão.

A MLM assume que os mapeamentos g_k são lineares, i.e. $g_k(\Phi_{n,\cdot}) = \Phi_{n,\cdot} B_{\cdot,k}$, em que B é uma matriz de coeficientes e corresponde aos parâmetros da MLM. Isso leva à função associada ao modelo MLM $h_B(x) : \mathcal{X} \to \mathcal{Y}$ dada por

$$h_B(x) = \arg\min_{y} \sum_{k=1}^{K} \left[\delta_k^2(y) - \left(\sum_{i=1}^{K} \phi_i(x) B_{i,k} \right)^2 \right]^2 \tag{1}$$

onde $\delta_k(y) = \|y - t_k\|$ representa a distância euclidiana entre y e o k-ésimo ponto de referência de saída $t_k \in \mathcal{Y}$; de forma similar, $\phi_i(x) = \|x - m_i\|$ denota a distância euclidiana entre x e o i-ésimo ponto de referência de entrada; K corresponde ao número de pontos de referência.

2.1. Algoritmo de aprendizagem

O algoritmo de aprendizagem da MLM requer a definição dos seguintes passos i) seleção do conjunto de referência $\{(m_k, t_k)\}$; e ii) determinação da matriz de parâmetros B. No que diz respeito à seleção de pontos de referência, na proposta original, o MLM extrai amostras aleatoriamente a partir do conjunto de dados disponíveis para a aprendizagem.

Uma vez que os pontos de referência são retirados dos dados, temos que $K \leq N$. O número de pontos de referências K controla a capacidade do modelo, portanto, pode ser usado para evitar *overfitting*. Sob circunstâncias normais onde o número de pontos de referência selecionados é menor do que o número de pontos de treinamento (i.e., $K < N$), a matriz B pode ser estimada via metódo de mínimos quadrados, i.e.,

$$\hat{B} = (\Phi^T \Phi)^{-1} \Phi^T \Delta, \tag{2}$$

onde Φ e Δ são matrizes de distância no espaço de entrada e saída, respectivamente.

2.2. Estimação da saída

A predição das saídas para novos dados de entrada refere-se principalmente à resolução do problema de minimização incorporado na Eq. (1). Para uma entrada

de teste x, cujas distâncias computadas para os K pontos de referência são dadas por $\phi_1(x) \dots \phi_K(x)$, podemos estimar as distâncias entre a saída desconhecida y e os pontos de referência (na saída) usando o modelo linear entre distâncias, isto é

$$\hat{\delta}_k(y) = \sum_{i=1}^{K} \phi_i(x) \hat{B}_{i,k}, \qquad \forall k = 1, \dots, K. \tag{3}$$

As estimativas $\hat{\delta}_1(y) \dots \hat{\delta}_K(y)$ podem então ser usadas para encontrar y no espaço de saída $\mathcal{Y}$. A localização de y pode ser estimada com a minização da Eq. (1) e reescrita aqui para enfatizar a dependência de y:

$$\hat{y} = \arg\min_{y} \sum_{k=1}^{K} \Big((y - t_k)^T (y - t_k) - \hat{\delta}_k^2(y) \Big)^2. \tag{4}$$

Vale mencionar que $\hat{\delta}_k(y)$ não é em uma função de y, mas sim, uma estimativa pontual da função de distância real $\delta_k(y) = \|y - t_k\|_2$. Assim, da perspectiva de otimização, $\hat{\delta}_k(y)$ deve ser tratada como constante. A intuição por trás do problema de estimar a saída desejada a partir de distâncias pode ser vista na Figura 1.

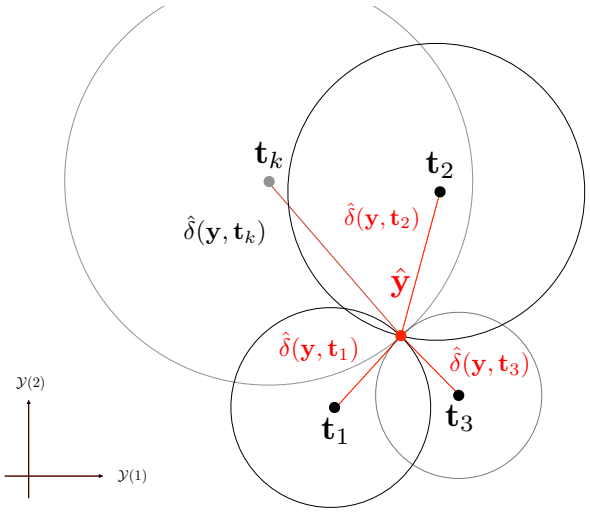

Figura 1. Estimação da saída.

A minimização da Eq. (4) também pode ser vista como um problema *multilateration*, uma vez que estamos interessados em localizar um ponto a partir das distâncias estimadas até os pontos de referência. A MLM original emprega o método de Levenberg-Marquadt (LM) para fornecer uma predição $\hat{y} = \arg\min_y J(y)$.

2.3. Pseudocódigo e Complexidade Computacional

O número de pontos de referência K é o único hyperparâmetro da MLM. Como usual, um valor para K pode ser encontrado através de técnicas de seleção de modelos, tal como validação cruzada.

O procedimento de treinamento da máquina de aprendizagem mínima está representado no algoritmo 1. O treinamento pode ser dividido basicamente em duas

partes: *i*) computar as matrizes de distâncias; *ii*) cálculo da solução de mínimos quadrados. A primeira tem custo $\Theta(KN)$, enquanto a segunda etapa tem custo $\Theta(K^2N)$, dado que a matriz pseudoinversa é encontrada utilizando SVD.

Algorithm 1 Treinamento da MLM

Require: Conjuntos de treinamento X, Y e K.
Ensure: $\hat{B}$, R e T.

1. Selecione aleatoriamente K pontos de referência, R, de X e suas saídas correspondentes, T, de Y;
2. Calcule Φ: Matriz de distâncias entre X e R;
3. Calcule Δ: Matriz de distâncias entre Y e T;
4. Calcule $\hat{B} = (\Phi^T\Phi)^{-1}\Phi^T\Delta$.

O procedimento de teste da máquina de aprendizagem mínima está representado no algoritmo 2.

Algorithm 2 Etapa de teste da MLM

Require: $\hat{B}$, R, T e x.
Ensure: $\hat{y}$.

1. Calcule $\phi_k(x) \quad \forall k = 1, \ldots, K$;
2. Calcule $\hat{\delta}_k(y) = \sum_{i=1}^{K} \phi_i(x)\hat{B}_{i,k}, \qquad \forall k = 1, \ldots, K.$
3. Use T e $\hat{\delta}_k(y)$ para encontrar uma estimativa $\hat{y}$ para a saída desejada.

Com relação à análise computacional da etapa de teste da MLM, consideraremos o método de Levenberg-Marquardt. Dessa forma, o custo computacional é dado por $\Theta(I(KS^2 + S^3))$, onde S é a dimensionalidade de y e I denota o número de iterações.

3. Máquina de Aprendizagem Mínima com Programação Linear

Nesta seção será apresentada a proposta deste trabalho, a máquina de aprendizagem mínima com programação linear (*Linear Programming based MLM*, LPMLM). A LPMLM propõe uma abordagem de programação linear para resolução do problema definido na Eq. (4). Assim, a etapa de treinamento da LPMLM é igual à da MLM.

A função custo dada na Eq. (4) é não-convexa. Dessa forma, mínimos globais não são garantidos. Este trabalho propõe, então reformular a função de custo da decisão da MLM, substituindo por um problema de programação linear em que a propriedade de convexidade é garantida.

Para isso, os termos quadráticos são removidos e o termo quadrático mais externo que mede o desvio em relação às distâncias na saída é substituído por valor absoluto, fornecendo então a função da LPMLM:

$$h_B^{PL}(x) = \arg\min_y \sum_{k=1}^{K} \left| |y - t_k| - \left(\sum_{i=1}^{K} \phi_i(x) B_{i,k} \right) \right| \tag{5}$$

É importante ressaltar que a formulação dada na Eq. (5) é válida somente para saídas unidimensionais, i.e., $S = 1$ e assim $|y - t_k|$ representa um escalar. Dessa forma, a aplicabilidade da LPMLM se dá a problemas de regressão.

O problema agora precisa ser colocado na forma de um programa linear, ou seja. Para isso, utilizamos o fato que o valor absoluto de $|y - t_k|$ é o menor valor z tal que $z \geq y - t_k$ e $z \geq -y + t_k$. Aplicando essa ideia na Eq. (5), chega-se a:

$$\begin{aligned}
\min \quad & \sum_{k=1}^{K} w_k \\
s.t. \quad & z_k \geq y - t_k, \quad k = 1, \cdots, K \\
& z_k \geq -y + t_k, \quad k = 1, \cdots, K \\
& w_k \geq z_k - \left(\sum_{i=1}^{K} \phi_i(x) B_{i,k} \right), \quad k = 1, \cdots, K \\
& w_k \geq -z_k + \left(\sum_{i=1}^{K} \phi_i(x) B_{i,k} \right), \quad k = 1, \cdots, K
\end{aligned}$$

As variáveis de decisão agora são $y, \{z_k\}_{k=1}^{K}, \{w_k\}_{k=1}^{K}$. Juntando as variáveis de decisão no vetor $p = [w_1, w_2, \ldots, w_K, z_1, \ldots, z_K, y]^T$, pode-se colocar o problema no formato $\min_p \{c^T p \mid Ap \geq b\}$, com

$$A = \begin{bmatrix} 0_{K\times K} & I_{K\times K} & -1_{K\times 1} \\ 0_{K\times K} & I_{K\times K} & 1_{K\times 1} \\ I_{K\times K} & -I_{K\times K} & 0_{K\times 1} \\ I_{K\times K} & I_{K\times K} & 0_{K\times 1} \end{bmatrix} \; b = \begin{bmatrix} -t_1 \\ \vdots \\ -t_K \\ t_1 \\ \vdots \\ t_K \\ -\sum_{i=1}^{K} \phi_i(x) B_{i,1} \\ \vdots \\ -\sum_{i=1}^{K} \phi_i(x) B_{i,K} \\ \sum_{i=1}^{K} \phi_i(x) B_{i,1} \\ \vdots \\ \sum_{i=1}^{K} \phi_i(x) B_{i,K} \end{bmatrix} \; c = \begin{bmatrix} 1_{k\times 1} \\ 0_{k\times 1} \\ 0_{1\times 1} \end{bmatrix}$$

O problema na forma geral está pronto para ser utilizado nas principais bibliotecas de programação linear. Neste trabalho, foi utilizado o solver do método simplex do software Octave.

O método simplex é caracterizado da seguinte forma: se um problema de programação linear no formato padrão tiver uma solução ótima, existe uma solução viável básica que é ótima. O algoritmo é baseado neste conceito e procura por

uma solução ótima ao passar de uma solução básica viável para outra, ao longo das bordas do conjunto viável, sempre em uma direção que permita a redução do custo. Eventualmente, uma solução básica viável é atingida e nenhuma direção conduz a uma redução de custo. Neste ponto, a solução viável é ótima e o algoritmo termina [Bertsimas and Tsitsiklis 1997].

O algoritmo simplex utilizado para resolver problemas de programação linear possui complexidade exponencial no pior caso, mas mesmo assim, para muitas instâncias ele é bastante rápido e tem comportamento polinomial para vários problemas práticos, o que torna ele um dos poucos exemplos conhecidos de algoritmo exponencial que é eficiente. O conjunto de problemas para os quais o simplex apresenta comportamento exponencial é muito pequeno.

4. Resultados

Nesta seção, são apresentados os resultados alcançados pela máquina de aprendizagem mínima usando programação linear em 5 problemas reais de regressão extraídos do repositório de aprendizagem de máquinas da University of California at Irvine (UCI Machine Learning — www.ics.uci.edu/ mlearn/). Todos os experimentos foram realizados utilizando o software Octave em plataforma Linux. Os conjuntos de dados são descritos na Tabela 1.

Tabela 1. Descrição dos conjuntos de dados: Dimensões de entrada, e número de amostras de treino e teste.

Conjunto de Dados	Entrada	Treino	Teste
Auto Price	15	106	53
Breast Cancer	32	129	65
Boston	13	337	169
Servo	4	111	56
Stocks	9	633	317

Os conjuntos de dados foram escolhidos para heterogeneidade do objeto no número de amostras de entradas. Todos os conjuntos de dados foram pré-processados da mesma maneira. As variáveis categóricas foram removidas bem como as amostras contendo valores faltantes. Dez diferentes permutações aleatórias dos conjuntos de dados inteiros são tomadas, e dois terços são usados para criar o conjunto de treinamento e o restante para o conjunto de teste. Em seguida, o conjunto de treinamento é normalizado com média zero e variância unitária e o conjunto de teste é normalizado usando a mesma média e variância do conjunto de treinamento.

O hiperparâmetro da Máquina de Aprendizagem Mínima, o número de pontos de referência (K), foi otimizado usando validação cruzada de 10 vezes (*10-fold crossvalidation*) com pontos de referência selecionados aleatoriamente em um intervalo de 5% a 100% (com um tamanho de passo de 5%) das amostra de treinamento disponíveis.

Todos os experimentos seguem a mesma diretriz encontrada em [SOUZA JUNIOR et al. 2013]. Todos os experimentos são repetidos por dez ro-

dadas independentes. Os modelos são comparados utilizando as estatísticas de média e desvio-padrão que são extraídas a partir do erro quadrático médio (*Mean Squared Error*, MSE) dos cinco conjuntos de dados. Os resultados estão disponíveis na Tabela 2. A LPMLM é comparada com a MLM original e outros dois métodos clássicos em aprendizagem de máquina: Máquina de Vetores Suporte para Regressão (*Support Vector Regression*, SVR) e Redes Perceptron multicamadas (*Multilayer Perceptrons*, MLP).

Tabela 2. Erro Quadrático Médio: média e desvio-padrão.

Conjunto de Dados	Métodos			
	LPMLM	MLM	MLP	SVR
Auto Price	$7.9e+6$	$2.6e+7$	$1.0e+7$	$9.8e+7$
	$3.8e+6$	$2.7e+7$	$3.9e+6$	$8.4e+6$
Breast Cancer	$1.1e+3$	$1.1e+3$	$1.5e+3$	$1.2e+3$
	$1.5e+2$	$1.8e+2$	$4.4e+3$	$7.2e+1$
Boston	$1.9e+1$	$1.9e+1$	$2.2e+1$	$3.4e+1$
	5.9	9.0	8.8	$3.1e+1$
Servo	$5.4e-1$	$4.6e-1$	$6.0e-1$	$6.9e-1$
	$3.0e-1$	$3.0e-1$	$3.2e-1$	$3.2e-1$
Stocks	$3.5e+1$	$4.1e-1$	$8.8e-1$	$5.1e-1$
	$4.6e+1$	$5.8e-2$	$2.1e-1$	$9.8e-2$

Pode ser observado que o modelo LPMLM apresenta resultado competitivo com todos os algoritmos em três conjuntos dentre os cinco avaliados. No conjunto Servo, o LPMLM não teve melhor desempenho que a MLM original, mas teve resultado superior ao MLP e SVR. Já no conjunto Stocks ele se mostrou menos eficiente que os três modelos. Além disso, o método LPMLM apresenta sempre a menor variância, mostrando assim maior confiabilidade quanto ao desempenho preditivo.

5. Conclusão

O artigo apresentou uma nova formulação para a etapa de teste da máquina de aprendizagem mínima. Tal formulação consiste de um problema linear que equivale a uma otimização (maximização ou minimização) de uma função linear sujeita a restrições lineares, que por sua vez possui mínimo global garantido. Isso evita então o aspecto variável de minimizar uma função custo não-convexa através de métodos baseados em gradiente, que é apresentado na MLM original.

Os resultados apresentados demonstraram a viabilidade da proposta, uma vez que a LPMLM mostrou desempenho no mínimo equivalente àquele alcançado via método MLM original com base em experimentos realizados com conjuntos de dados reais do UCI Repository (problemas de regressão). Além disso, a formulação proposta permitiu reduzir a variância nos resultados alcançados pela MLM.

6. Referências

Bertsimas, D. and Tsitsiklis, J. N. (1997). *Introduction to Linear Optimization*.

Bishop, C. M. et al. (2006). *Pattern recognition and machine learning*, volume 1. Springer New York.

Hastie, T., Tibshirani, R., and Friedman, J. (2001). *Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer.

Hinton, G. E. and Salakhutdinov, R. R. (2006). Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507.

Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q., editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc.

Lenz, I., Lee, H., and Saxena, A. (2015). Deep learning for detecting robotic grasps. *International Journal of Robotics Research (IJRR) Special Issue on Robot Vision*.

Sainath, T., Kingsbury, B., Saon, G., Soltau, H., rahman Mohamed, A., Dahl, G., and Ramabhadran, B. (2015). Deep convolutional neural networks for large-scale speech tasks. *Neural Networks*, 64:39–48.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503.

SOUZA JUNIOR, A. H., Corona, F., Miché, Y., Lendasse, A., Barreto, G., and Simula, O. (2013). Minimal learning machine: A new distance-based method for supervised learning. In *Proceedings of the 12th International Work Conference on Artificial Neural Networks (IWANN'2013)*, volume 7902 of *Lecture Notes in Computer Science*, pages 408–416. Springer.

Souza Júnior, A. H., Corona, F., Barreto, G. A., Miche, Y., and Lendasse, A. (2015). Minimal learning machine: A novel supervised distance-based approach for regression and classification. *Neurocomputing*, 164:34 – 44.