TÓPICOS DE ÁLGEBRA LINEAR E O TEOREMA DE CONVERGÊNCIA DA REDE NEURAL PERCEPTRON

Data de aceite: 03/04/2023

David Hapner Barzotto
Universidade Estadual do Oeste do Paraná

Simone Aparecida Miloca
Universidade Estadual do Oeste do Paraná

Agradecemos a Fundação Araucaria pela bolsa no projeto PIBIC.

RESUMO: A Álgebra Linear tem uma aplicabilidade em um vasto campo das ciências puras e aplicadas, como matemática, física, computação e as engenharias. Este trabalho tem o intuito de mostrar resultados algébricos utilizados na demonstração do teorema de convergência da primeira grande Rede Neural Artificial (RNA), o Perceptron. Mostraremos que o Perceptron sempre irá encontrar um hiperplano separador para a classificação binária de dados.

PALAVRAS-CHAVE: Álgebra Linear; Perceptron; Redes Neurais Artificiais (RNAs).

1 | INTRODUÇÃO

As Redes Neurais Artificiais (RNAs) fazem parte do campo de estudo da Inteligência Artificial (IA), porém, também estão presentes em outras áreas do conhecimento tais como computação evolucionária, metaheurística e machine learning, sendo capazes de resolver os mais diversos problemas, como classificação, regressão e clusterização. Possuem inspiração nas Redes Neurais Biológicas (RNBs) e seus principais filamentos, como o dendrito, o corpo celular e o axônio. A Figura 1 exemplifica um modelo biológico e um artificial com neurônios e conexões direcionadas entre eles. Devido esta motivação biológica, os elementos de processamento de uma rede neural são denominados neurônios ou nós, sendo uma unidade de processamento de informações fundamentais para o funcionamento da rede. A partir disso, pode-se definir três conceitos básicos que caracterizam uma rede neural artificial:

- Um padrão da rede chamado

de pesos sinápticos, também conhecidos por conexões sinápticas que, unidos com uma operação binária, combinam os dados de entrada pertencentes a um vetor **x** com um vetor dos pesos sinápticos **w** montando um hiperplano separador das classes dos dados na forma de $\mathbf{w}^T\mathbf{x} = 0$;

- Uma regra de agregação de dados que correlaciona as entradas dadas aos neurônios, ponderados com suas respectivas conexões sinápticas;
- Uma função chamada de ativação com o intuito de modelar o cálculo dos dados podendo gerar funções não lineares ou condicionar a saída do neurônio em um intervalo determinado.

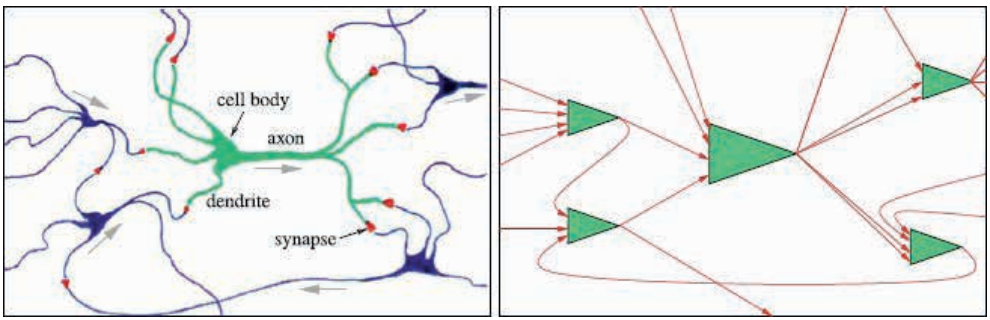

Figura 1: Modelo biológico e modelo artificial de uma rede neural.

Fonte: Extraído de Ertel (2017).

A Figura 2 exemplifica o modelo de um neurônio. Tem-se X_1, X_2, ..., X_m sinais de en- trada conectados ao neurônio k que são multiplicados pelos pesos sinápticos, W_{k1}, W_{k2}, ..., W_{km}, incluindo uma variável polarização (bias) b_k que é adicionada ao somatório da função de ativação φ, com o intuito de aumentar o grau de liberdade desta função e, consequentemente, a capacidade de aproximação da rede.

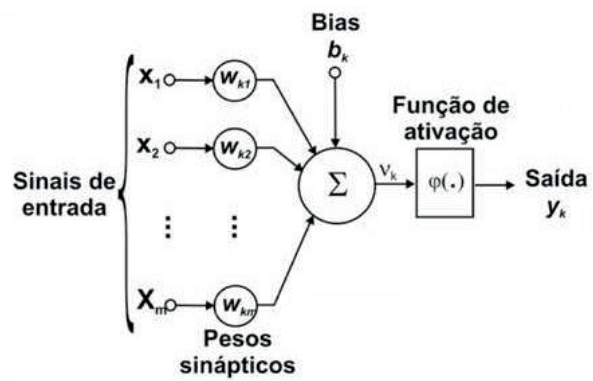

Figura 2: Modelo de um Neurônio Artificial.

Fonte: Extraído de Haykin (2001).

Matematicamente, um neurônio k é escrito como

$$y_k = \varphi\left(\sum_{j=1}^{m} W_{kj}\, X_j + b_k\right)$$

onde, $\varphi(.)$ é a função de ativação e y_k são os sinais de saída do neurônio.

Segundo Haykin (2001), o Perceptron é a rede neural mais simples. Foi construído com o foco de solucionar classificações binárias e é uma rede considerada linear, de uma camada apenas, ou seja, supondo que se tenha um conjunto de entradas e se conheça suas respectivas saídas, o Perceptron deve aprender os padrões e classificar corretamente novas entradas. Este trabalho tem por objetivo apresentar o teorema de convergência do Perceptron com a adaptação baseada em Haykin (2001), escrevendo sobre os principais conceitos de Álgebra Linear necessários para a prova do teorema.

2 | TÓPICOS DE ÁLGEBRA LINEAR

Esta seção tem por objetivo escrever conceitos e resultados de Álgebra Linear utilizados na parte que envolve o Teorema de Convergência do Perceptron.

O texto que segue refere-se a espaços vetoriais reais. Temos interesse particular no espaço vetorial euclidiano, que é um espaço vetorial real, de dimensão finita, em que está definido um produto interno. As principais referências foram Poole (2003) e Steinbruch e Winterle (1995).

2.1 Dependência Linear em Espaços Vetoriais

Definição 1. Sejam $\{v_1, v_2, v_3, ..., v_n\}$ um conjunto de vetores de um espaço vetorial V e os escalares $\alpha_1, \alpha_2, \alpha_3, ..., \alpha_n$, temos que:

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 + \cdots + \alpha_n v_n$$

é dito uma combinação linear dos vetores $v_1, v_2, v_3, ..., v_n$.

Exemplo 1. Sejam $\mathbf{v}_1 = (1, 5)$ e $\mathbf{v}_2 = (4, -1)$. Então $\mathbf{v} = (14, 7)$ é uma combinação linear de $\mathbf{v}_1$ e $\mathbf{v}_2$ pois $\mathbf{v} = (14, 7) = 2\mathbf{v}_1 + 3\mathbf{v}_2 = 2(1, 5) + 3(4, -1)$.

A definição a seguir apresenta o conceito de dependência e independência linear entre vetores.

Definição 2. Seja V um espaço vetorial e $S = \{v_1, v_2, ..., v_n\} \subset V$. S é dito linearmente dependente (LD) se existirem escalares $\alpha_1, \alpha_2, ..., \alpha_n$, com pelo menos um $\alpha_i \neq 0$ tais que

$$\alpha_1 v_1 + \cdots + \alpha_n v_n = 0.$$

S é dito linearmente independente (LI) se não for linearmente dependente.

2.2 Produto Interno e Norma

Definição 3. Seja *V* um espaço vetorial real. Um produto interno é uma função

$$\langle\ ,\ \rangle : V \times V \to \mathbb{R}$$
$$(u,\ v) \to \langle u,\ v\rangle$$

que satisfaz as propriedades, para todos *u*, *v*, *w* em *V* e $\alpha \in \mathbb{R}$

i) $\langle u,\ u\rangle \geq 0$, e $\langle u,\ u\rangle = 0 \Leftrightarrow u = 0$

ii) $\langle au,\ v\rangle = \alpha\ \langle u,\ v\rangle$

iii) $\langle u + v,\ w\rangle = \langle u,\ w\rangle + \langle v,\ w\rangle$

iv) $\langle u,\ v\rangle = \langle v,\ u\rangle$

O Exemplo 2 é um produto interno, também chamado de produto escalar usual de vetores em $\mathbb{R}^n$.

Exemplo 2. Sejam $\mathbf{u} = (x_1,\ x_2,\ \ldots,\ x_n)$ *e* $\mathbf{v} = (y_1,\ y_2,\ \ldots,\ y_n)$. Tem-se

$$\langle \mathbf{u},\ \mathbf{v}\rangle = (x_1y_1 + x_2y_2 + \cdots + x_ny_n) = \mathbf{u}\mathbf{v}^T\ .$$

Em um espaço vetorial *V* com produto interno, podemos definir norma de um vetor *v* $\in$ *V* que provém deste produto interno como

$$\|v\| = \sqrt{\langle v, v\rangle}. \qquad (1)$$

Exemplo 3. Em $\mathbb{R}^2$, seja $\mathbf{v} = (x_1,\ x_2)$. A norma é dada por,

$$\|\mathbf{v}\| = \sqrt{\langle \mathbf{v},\mathbf{v}\rangle} = \sqrt{x_1^2 + x_2^2}.$$

Assim, temos que dado um vetor $\mathbf{v} = (x_1,\ x_2,\ x_3,\ \ldots,\ x_n)$ em $\mathbb{R}^n$, a norma euclidiana é definida por

$$\|\mathbf{v}\| = \sqrt{\langle \mathbf{v},\mathbf{v}\rangle} = \sqrt{x_1^2 + x_2^2 + x_3^2 + \ldots + x_n^2}.$$

2.3 Desigualdade de Cauchy-Schwarz

Teorema 4. *Seja V um espaço vetorial com produto interno, então:*

$$|\ \langle u,v\rangle\ | \leq ||u||\ ||v||,\ \forall u,v \in V.$$

Prova. Os vetores *u* e *v* serão LD ou LI. Faremos a prova considerando cada uma destas situações. Caso 1. Sejam *u* e *v* linearmente dependentes, logo temos $\alpha \in$ R tal que $u = \alpha v$. Então,

$$|\langle u,v\rangle| = |\langle \alpha v, v\rangle| = |\alpha|\langle v,v\rangle = |\alpha|\left(\sqrt{\langle v,v\rangle}\right)^2 = |\alpha|\sqrt{\langle v,v\rangle}\sqrt{\langle v,v\rangle}$$

usando a propriedade (ii) descrita na Definição 3 e a igualdade (1), temos,

$$|\alpha|\sqrt{\langle v,v\rangle}\sqrt{\langle v,v\rangle} = \sqrt{\alpha^2\langle v,v\rangle}\sqrt{\langle v,v\rangle} = \sqrt{\langle \alpha v,\alpha v\rangle}\sqrt{\langle v,v\rangle} = \sqrt{\langle u,u\rangle}\sqrt{\langle v,v\rangle} = ||u||\,||v||.$$

Segue que vale a igualdade,

$$|\ \langle u,v\rangle\ | = ||u||\,||v||.$$

Caso 2. Sejam *u* e *v* linearmente independentes, logo, existe α ∈ ℝ em que *u*+α*v*≠0, então $\langle u + \alpha v, u + \alpha v\rangle > 0$, com isso,

$$\langle u + \alpha v, u + \alpha v\rangle = \langle u,u\rangle + \langle u,\alpha v\rangle + \langle \alpha u, v\rangle + \langle \alpha v,\alpha v\rangle > 0.$$

Pela Definição 3 teremos

$$\langle u,u\rangle + \alpha\langle u,v\rangle + \alpha\langle u,v\rangle + \alpha^2\langle v,v\rangle = \langle u,u\rangle + 2\alpha\langle u,v\rangle + \alpha^2\langle v,v\rangle > 0.$$

A inequação de grau 2 está em função de α e não apresenta raızes reais, apenas complexas, o seu discriminante Δ deve ser menor que 0, ou seja,

$$\Delta < 0 \Rightarrow (2\langle u,v\rangle)^2 - 4(\langle u,u\rangle)(\langle v,v\rangle) < 0 \Rightarrow 4\langle u,v\rangle^2 < 4(\langle u,u\rangle)(\langle v,v\rangle) \Rightarrow \langle u,v\rangle^2 < \langle u,u\rangle\langle v,v\rangle$$

podemos extrair a raiz quadrada da inequação,

$$\sqrt{(\langle u,v\rangle^2)} < \sqrt{\langle u,u\rangle}\sqrt{\langle v,v\rangle} \Rightarrow \langle u,v\rangle < \sqrt{\langle u,u\rangle}\sqrt{\langle v,v\rangle}.$$

Por (1) temos,

$$|\ \langle u,v\rangle\ | < ||u||\,||v||.$$

Pelo Caso 1 e Caso 2 temos a validade da desigualdade de Cauchy-Schwarz,

$$|\ \langle u,v\rangle\ | \leq ||u||\,||v||.$$

2.4 Hiperplanos

Definição 5. Um hiperplano *H* do $\mathbb{R}^n$ é o conjunto

$$H = \{(x_1, x_2, ..., x_n) \in \mathbb{R}^n \mid a_1x_1 + a_2x_2 + \cdots + a_nx_n = b\ ;\ a_1, \ldots, a_n \in \mathbb{R}\}.$$

As Figuras 3 e 4 exemplificam hiperplanos em $\mathbb{R}^2$ (retas).

Um hiperplano divide $\mathbb{R}^n$ em dois semiespaços H_1 e H_2.

$$H_1 = \{(x_1, x_2, ..., x_n) \in \mathbb{R}^n \mid a_1x_1 + a_2x_2 + \cdots + a_nx_n \geq b\ ;\ a_1, \ldots, a_n \in \mathbb{R}\}$$

$$H_2 = \{(x_1, x_2, ..., x_n) \in \mathbb{R}^n \mid a_1x_1 + a_2x_2 + \cdots + a_nx_n \leq b\ ;\ a_1, \ldots, a_n \in \mathbb{R}\}.$$

Um hiperplano pode ser escrito na notação de produto escalar, como

$$\mathbf{a}^T\mathbf{x} = 0$$

onde $\mathbf{a} = (b, a_1, ..., a_m)^T$ e $\mathbf{x} = (1, x_1, ..., x_m)^T$.

A próxima definição traz o conceito de conjuntos linearmente separáveis.

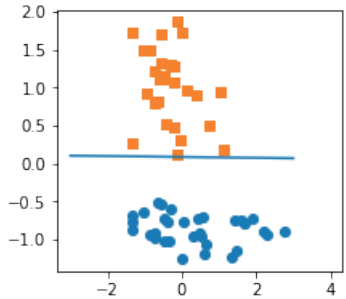

Figura 3: Exemplo de Hiperplano.

Fonte: Produzido pelo autor.

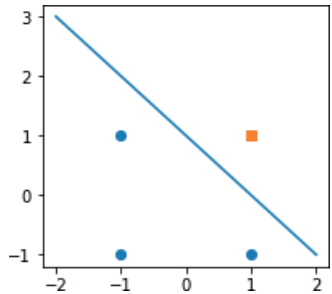

Figura 4: Hiperplano para a função AND booleana.

Fonte: Produzido pelo autor.

Definição 6. Dois conjuntos $C_1 \subset \mathbb{R}^{m+1}$ e $C_2 \subset \mathbb{R}^{m+1}$ são linearmente separáveis se existir $\mathbf{a} = (b, a_1, ..., a_m)^T$, onde $b, a_1, ..., a_m$ são números reais, tais que

$$\mathbf{a}^T \mathbf{x} > 0, \forall \mathbf{x} = [1, x_1, ..., x_m]^T \in C_1$$

e

$$\mathbf{a}^T \mathbf{x} \leq 0, \forall \mathbf{x} = [1, x_1, ..., x_m]^T \in C_2$$

em que b é o coeficiente linear do hiperplano, que tem o nome de bias quando se trata de hiperplanos de RNAs.

3 | REDE NEURAL PERCEPTRON

Formalmente descrito por Fausett (1994), Haykin (2001) e Kovács (2006), o Perceptron, criado e idealizado por Frank Rosenblatt, segue os parâmetros da álgebra booleana, ou seja, AND e OR, logo um senso binário {0,+1} ou bipolar {-1,+1}. No caso do Perceptron, seu algoritmo iterativo de ajuste de pesos é calculado como mostra o Algoritmo 1. A ideia principal do Perceptron é classificar duas classes linearmente separáveis, C_1 e C_2. Consequentemente, tal classificação só é possível pois a rede gera um hiperplano separador em m-dimensões descrito como $\sum_{i=1}^{m} wi(n)xi(n) + b(n) = 0$ ou na forma de produto interno $\mathbf{w}^T(n)\mathbf{x}(n) = 0$, em que n é o número de iterações feitas. O funcionamento se dá pela seguinte forma:

[1] Caso o n-ésimo termo de **x** seja classificado corretamente na n-ésima iteração, então o vetor **w** não é corrigido.

$$\mathbf{w}(n+1) = \mathbf{w}(n), \text{ se } \mathbf{w}^T \mathbf{x} > 0. \text{ Logo } \mathbf{x}(n) \in C_1 \quad (2)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n), \text{ se } \mathbf{w}^T \mathbf{x} \leq 0. \text{ Logo } \mathbf{x}(n) \in C_2. \quad (3)$$

[2] Se o passo 1 for falso, faz-se as seguintes atualizações.

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \eta(n)\mathbf{x}(n), \text{ se } \mathbf{w}^T\mathbf{x} > 0. \text{ Logo } \mathbf{x}(n) \square C_2 \quad (4)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \eta(n)\mathbf{x}(n), \text{ se } \mathbf{w}^T\mathbf{x} \leq 0. \text{ Logo } \mathbf{x}(n) \square C_1 \quad (5)$$

onde η é um parâmetro denominado taxa de aprendizado e geralmente é um valor entre zero e um.

Entrada: Inicializar o vetor de pesos **w** e o bias *b*;
Inicie a taxa de aprendizado η ($0 < \eta \leq 1$);
while *Critério de Parada não satisfeito* **do**
for *Para cada par de dados de treinamento* (x_i, d_i) **do**
Execute ;

$$y = b + \sum_{i=1}^{n} w_i x_i$$

if $y = d$ **then**
ocorre [**1**] ;
$w_i(novo) = w_i(atual)$;
$b(novo) = b(atual)$;
else
ocorre [**2**] ;
Atualizar os pesos e a tendência;
$w_i(novo) = w_i(atual) + \eta x_i$; $b(novo) = b(atual) + \eta$;
end end
Teste a condição de parada.
end
Algoritmo 1: Algoritmo *Perceptron*

3.1 Exemplo de Funcionamento do Perceptron

Exemplo 4. Utilize a rede neural Perceptron para classificar a função lógica "AND", cujas entradas e saídas são mostradas na Tabela 1. Considere a taxa de aprendizado η como sendo fixo e igual a 1, o valor de entrada para o bias também 1, com peso $b(n) = 0$.

	Entrada		Saída
	$x_1(n)$	$x_2(n)$	d_i
E1	0	0	0
E2	0	1	0
E3	1	0	0
E4	1	1	1

Tabela 1: Tabela para a função AND

Fonte: Produzido pelo autor.

Primeira Iteração

1. Primeiro padrão E1 é apresentado à rede.

$$y(1) = b + \sum_{i=1}^{2} w_i(1)x_i(1) = 0 + 0 \times 0 + 0 \times 0 = 0, \text{ assim, } d(1) = y(1) = 0.$$

Como $d(1) = y(1)$, os pesos não se atualizam, logo,

$$w_1(2) = w_1(1) = 0,\ w_2(2) = w_2(1) = 0,\ b(2) = b(1) = 0.$$

2. Segundo padrão E2 é apresentado à rede.

$$y(2) = b(2) + \sum_{i=1}^{2} w_i(2)x_i(2) = 0 + 0 \times 0 + 0 \times 1 = 0. \text{ Assim, } d(2) = y(2) = 0$$

Como $d(2) = y(2)$, os pesos não se atualizam, assim,

$$w_1(3) = w_1(2) = 0,\ w_2(3) = w_2(2) = 0,\ b(3) = b(2) = 0.$$

3. Terceiro padrão E3 é apresentado à rede.

$$y(3) = b(3) + \sum_{i=1}^{2} w_i(3)x_i(3) = 0 + 0 \times 1 + 0 \times 0 = 0, \text{ assim, } d(3) = y(3) = 0.$$

Como $d(3) = y(3)$, os pesos não se atualizam, logo,

$$w_1(4) = w_1(3) = 0,\ w_2(4) = w_2(3) = 0,\ b(4) = b(3) = 0..$$

4. Quarto padrão E4 é apresentado à rede.

$$y(4) = b(4) + \sum_{i=1}^{2} w_i(4)x_i(4) = 0 + 0 \times 1 + 0 \times 1 = 0, \text{ logo, } d(4) \neq y(4).$$

Como $d(4) \neq y(4)$, os pesos são atualizados, logo temos,

$$w_1(5) = w_1(4) + \eta(4)x_1(4) = 0 + 1 \times 1 = 1$$

$$w_2(5) = w_2(4) + \eta(4)x_2(4) = 0 + 1 \times 1 = 1$$

$$b(5) = b(4) + \eta(4) \times 1 = 0 + 1 \times 1 = 1.$$

A Figura 5 mostra os resultados da primeira iteração. Após 10 épocas (iterações) tem-se o hiperplano separador que é mostrado na Figura 6, cuja equação final fica $2x_1+x_2-2=0$.

$d(n)$	$x_1(n)$	$x_2(n)$	$w_1(n)$	$w_2(n)$	$bias$
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
1	1	1	0	0	0
-	-	-	1	1	1

Figura 5: 1ª Iteração do Perceptron.

Fonte: Produzido pelo autor.

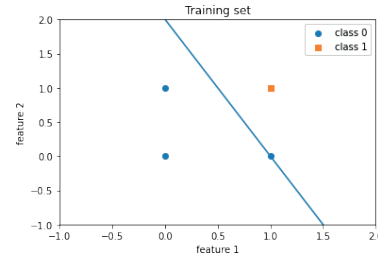

Figura 6: Hiperplano Perceptron - função AND.

Fonte: Produzido pelo autor.

4 | TEOREMA DE CONVERGÊNCIA DO PERCEPTRON

Teorema 7. *Sejam* **x** *um vetor de entrada,* **w** *o vetor peso, 1 o valor de entrada para o bias e n o passo por iteração. Tome as classes* C_1 *e* C_2 *linearmente separáveis por um hiperplano* $\mathbf{w}^T\mathbf{x} = 0$*. Então o algoritmo Perceptron converge para um* n_{max} *de iterações para qualquer inicialização do vetor de pesos* **w***, sem qualquer critério de parada preestabelecido, com isso gerando um hiperplano separador através de um* n_{max}*:*

$$n_{max} = \frac{\beta\|\mathbf{w}_0\|^2}{\alpha^2}.$$

Nossa notação considera

$\mathbf{x}(n) = [1, x_1(n), x_2(n), x_3(n), ..., x_m(n)]^T$ e $\mathbf{w}(n) = [b(n), w_1(n), w_2(n), w_3(n), ..., w_m(n)]^T$

onde n é a n-ésima iteração e a saída da combinação linear é dada por

$$y(n) = \sum_{i=0}^{m} w_i(n)\, x_i(n) = \mathbf{w}^T(n)\mathbf{x}(n).$$

Definimos $w_0(n) = b(n)$. Para n fixo, $\mathbf{w}^T\mathbf{x} = 0$ descreve um hiperplano separador de duas classes C_1 e C_2. Como os dados de entrada das duas classes C_1 e C_2 são linearmente separáveis, então existe um subconjunto de vetores de treinamento $\mathbf{x}_1(n) = X_1 \in C_1$ e outro subconjunto $\mathbf{x}_2(n) = X_2 \in C_2$ sendo $X_1 \cup X_2 = X$, o conjunto X de treinamento completo. Dados tais vetores de entrada, os pesos serão ajustados de modo que as classes sejam linearmente separáveis,

$$\mathbf{w}^T\mathbf{x} > 0, \forall x \in C_1 \qquad e \qquad \mathbf{w}^T\mathbf{x} \leq 0, \forall x \in C_2.$$

Prova. Seja $\eta(n) = \eta > 0$, em que η é uma constante independente do número de iterações. Tomemos $\eta = 1$, para termos a regra de adaptação com incremento fixo para o Perceptron. Considere a condição inicial $\mathbf{w}(0) = \mathbf{0}$ e suponha que $\mathbf{w}^T(n)\mathbf{x}(n) < 0$ e $\mathbf{x}(n) \in X_1$. Ou seja, considere por hipótese que o Perceptron classifica incorretamente os vetores

de entrada **x**. Como a parte 1 do algoritmo de aprendizado é falsa, vamos para a segunda parte do Algoritmo de aprendizado, que é a a atualização dos pesos,

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{x}(n), \ \forall \mathbf{x}(n) \in C_1. \tag{6}$$

Dado a condição de inicio $\mathbf{w}(0) = \mathbf{0}$, temos que

$\mathbf{w}(1) = \mathbf{w}(0) = \mathbf{0}$

$\mathbf{w}(2) = \mathbf{w}(1) + \mathbf{x}(1)$

$\mathbf{w}(3) = \mathbf{w}(2) + \mathbf{x}(2) = \mathbf{w}(1) + \mathbf{x}(1) + \mathbf{x}(2) = \mathbf{0} + \mathbf{x}(1) + \mathbf{x}(2)$

assim, iterativamente temos,

$$\mathbf{w}(n+1) = \mathbf{x}(1) + \mathbf{x}(2) + \mathbf{x}(3) + \cdots + \mathbf{x}(n) = \sum_{i=1}^{n} \mathbf{x}(i). \tag{7}$$

Como assumimos que C_1 e C_2 são linearmente separáveis, então existe uma solução $\mathbf{w}_0$ de $\mathbf{w}^T\mathbf{x} > 0$ para $\mathbf{x}(n) \in X_1$. Assim, podemos definir um número positivo α, tal que

$$a = \min_{\mathbf{x}(n) \in X_1} \mathbf{w}_0^T \mathbf{x}(n). \tag{8}$$

Multiplicando ambos os lados da equação 7 por $\mathbf{w}_0^T$, temos,

$$\mathbf{w}_0^T \mathbf{w}(n+1) = \mathbf{w}_0^T \sum_{i=1}^{n} \mathbf{x}(i) \tag{9}$$

Pela equação 8 e a equação 9, tem-se a desigualdade

$$\mathbf{w}_0^T \mathbf{w}(n+1) \geq na. \tag{10}$$

Agora, utilizando a desigualdade de Cauchy-Schwarz (Teorema 4), para os vetores $\mathbf{w}_0$ e $\mathbf{w}(n+1)$, teremos a desigualdade,

$$||\mathbf{w}_0||\,||\mathbf{w}(n+1)|| \ \geq \ \mathbf{w}_0^T \mathbf{w}(n+1)$$

e elevando ao quadrado ambos os lados, encontramos,

$$||\mathbf{w}_0||^2\,||\mathbf{w}(n+1)||^2 \ \geq \ [\mathbf{w}_0^T \mathbf{w}(n+1)]^2 \tag{11}$$

Como vale a desigualdade 10, notando que ambos os membros são não negativos, teremos,

$$||\mathbf{w}_0||^2\,||\mathbf{w}(n+1)||^2 \ \geq \ [\mathbf{w}_0^T \mathbf{w}(n+1)]^2 \geq n^2\alpha^2.$$

Assim encontramos

$$||\mathbf{w}_0||^2\,||\mathbf{w}(n+1)||^2 \ \geq n^2\alpha^2.$$

$$\|\mathbf{w}(\boldsymbol{n}+\mathbf{1})\|^2 \ \geq \ \frac{n^2\alpha^2}{\|\mathbf{w}_0\|^2} \tag{12}$$

A inequação 12 nos diz que toda parte acima da curva da função quadrática é uma

possível solução mínima, como exemplifica a parte (**a**) da Figura 7.

Por outro lado, a partir da equação 6 temos que,

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{x}(k), \text{ para } k = 1, \ldots, n, \mathbf{x}(k) \in X_1. \quad (13)$$

Calculando a norma euclidiana temos

$$||\mathbf{w}(k+1)|| = ||\mathbf{w}(k) + \mathbf{x}(k)||. \quad (14)$$

Elevando ao quadrado ambos os lados da equação 14 temos que,

$$||\mathbf{w}(k+1)||^2 = ||\mathbf{w}(k) + \mathbf{x}(k)||^2 = ||\mathbf{w}(k)||^2 + 2\mathbf{w}^T(k)\mathbf{x}(k) + ||\mathbf{x}(k)||^2.$$

Da suposição que o Perceptron classifica erroneamente o vetor de entrada $\mathbf{x}(k) \in X_1$, temos $\mathbf{w}^T(k)\mathbf{x}(k) < 0$, logo, o fator $2\mathbf{w}^T(k)\mathbf{x}(k)$ é negativo, assim obtemos,

$$||\mathbf{w}(k+1)||^2 \leq ||\mathbf{w}(k)||^2 + ||\mathbf{x}(k)||^2, k = 1, 2, 3, \ldots, n$$

ou de forma equivalente,

$$||\mathbf{w}(k+1)||^2 - ||\mathbf{w}(k)||^2 \leq ||\mathbf{x}(k)||^2, k = 1, 2, 3, \ldots, n.$$

Como k é definido a partir de n e da hipótese $\mathbf{w}(0) = \mathbf{0}$, temos que existe $\beta > 0$ tal que

$$\|\mathbf{w}(n+1)\|^2 \leq \sum_{k=1}^{n} \|\mathbf{x}(k)\|^2 \leq n\beta$$

logo,

$$\|\mathbf{w}(n+1)\|^2 \leq n\beta. \quad (15)$$

A equação 15 nos mostra que a solução da curva linear é delimitada pela parte inferior da função, como é possível ver na parte (**b**) da Figura 7. Assim, define-se que β é um valor positivo dado pela equação 16

$$\beta = \max_{\mathbf{x}(k) \in X_1} ||\mathbf{x}(k)||^2. \quad (16)$$

Contudo, a inequação 15 é contrária a 12 para valores suficientemente grandes de n, como exemplifica a Figura 7. A inequação 15 diz que os pesos tendem a crescer infinitamente, o que de fato é impossível. A partir disso, inferimos que n não pode ser maior que um número máximo. Assim, ambas as inequações são satisfeitas com o sinal de igualdade, como ilustrado na parte (**c**) da Figura 7, em que o eixo das ordenadas é definido por $||\mathbf{w}(n)||^2$ e o eixo das abscissas é definido pelo número de iterações n da rede. Portanto,

$$\frac{n_{max}^2 \alpha^2}{\|\mathbf{w}_0\|^2} = n_{max}\beta \Leftrightarrow n_{max} = \frac{\beta \|\mathbf{w}_0\|^2}{\alpha^2}. \quad (17)$$

A prova foi feita para para $\eta(n) = 1$ pois facilita os cálculos, já que seu valor não é importante desde que seja positivo, e, caso $\eta \neq 1$, apenas escala os vetores padrões sem

afetar seu hiperplano separador. Existindo um vetor solução $\mathbf{w}_0$, a atualização dos pesos deve chegar a um valor máximo de iterações n_{max} para gerar um hiperplano para as classes C_1 e C_2. Provamos para η = 1. Como a inequação 12 cresce quadraticamente em função de n e a inequação 15 cresce linearmente em função de n, sempre existirá um n para qualquer α ou β pertencente aos reais, em que a sua intersecção satisfaça o critério de parada.

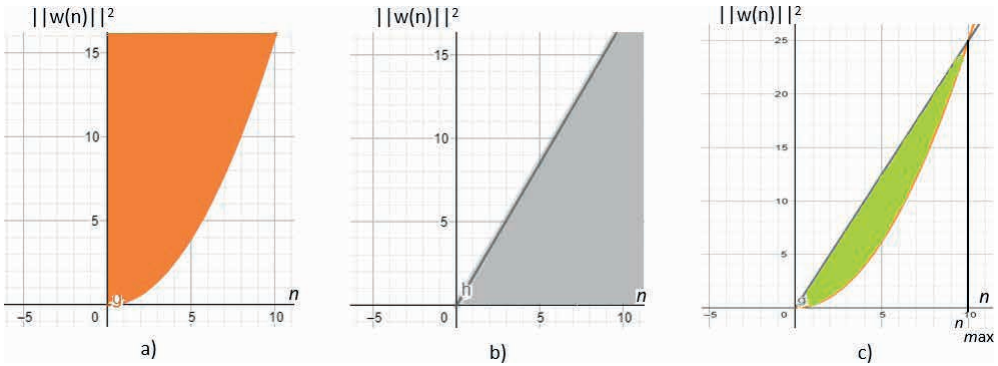

Figura 7: **a** solução da inequação 12; **b** solução da inequação 15; **c** solução da equação 17.

Fonte: Produzido pelo autor.

5 | CONCLUSÃO

Neste trabalho mostramos o teorema de convergência para o Perceptron. Percebemos a utilização de tópicos estudados na Álgebra Linear como a desigualdade de Cauchy-Schwarz, que ao ser definida e usada na prova, limita o número de iterações a um valor mínimo α e a norma euclidiana que limita a um valor máximo β.

REFERÊNCIAS

ERTEL, Wolfgang. **Introduction to Artificial Intelligence**. 2. ed. Londres: Springer, 2017. 356 p.

FAUSETT, Laurene. **Fundamentals of Neural Networks**: architectures, algorithms and applications. Nova Iorque: Prentice-Hall, 1994. 461 p.

HAYKIN, Simon. **Redes Neurais**: princípios e prática. 2. ed. São Paulo: Bookman, 2001. 900 p.

KOVACS, Zsolt L.. **Redes Neurais Artificiais**: fundamentos e aplicações. 4. ed. São Paulo: Livraria da Física, 2006. 171 p.

POOLE, David. **Álgebra Linear**. São Paulo: Cengage Ctp, 2003. 718 p.

STEINBRUCH, Alfredo; WINTERLE, Paulo. **Álgebra Linear**. São Paulo: Pearson Universi- dades, 1995. 600 p.