

**Título do trabalho a ser apresentado à
CPG para a dissertação/tese**

Aarão Melo Lopes

DISSERTAÇÃO/TESE APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
OBTENÇÃO DO TÍTULO
DE
MESTRE/DOCTOR EM CIÊNCIAS

Programa: Nome do Programa

Orientador: Prof. Dr. Nome do Orientador

Coorientador: Prof. Dr. Nome do Coorientador

Durante o desenvolvimento deste trabalho o autor recebeu auxílio financeiro da
CAPES/CNPq/FAPESP

São Paulo, fevereiro de 2011

Redes Neurais Convolucionais Quaternion

Esta é a versão original da dissertação elaborada pelo
candidato (Aarão Melo Lopes), tal como
submetida à Comissão Julgadora.

Título do trabalho a ser apresentado à CPG para a dissertação/tese

Esta versão da dissertação/tese contém as correções e alterações sugeridas pela Comissão Julgadora durante a defesa da versão original do trabalho, realizada em 14/12/2010. Uma cópia da versão original está disponível no Instituto de Matemática e Estatística da Universidade de São Paulo.

Comissão Julgadora:

- Prof^a. Dr^a. Nome Completo (orientadora) - IME-USP [sem ponto final]
- Prof. Dr. Nome Completo - IME-USP [sem ponto final]
- Prof. Dr. Nome Completo - IMPA [sem ponto final]

Agradecimientos

[illegible]

Resumo

SOBRENOME, A. B. C. **Título do trabalho em português**. 2010. 120 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2010.

[illegible]

Palavras-chave: palavra-chave1, palavra-chave2, palavra-chave3.

Abstract

SOBRENOME, A. B. C. **Título do trabalho em inglês**. 2010. 120 f. Tese (Doutorado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2010.

[illegible]

Keywords: keyword1, keyword2, keyword3.

Sumário

Lista de Abreviaturas	ix
Lista de Símbolos	xi
Lista de Figuras	xiii
Lista de Tabelas	xv
1 Introdução	1
1.1 Considerações Preliminares	1
1.2 Objetivos	1
1.3 Contribuições	2
1.4 Organização do Trabalho	2
2 Conceitos	3
2.1 Perceptrons de Camada Única	3
2.1.1 Teorema de convergência do perceptron	3
2.2 Perceptrons de Múltiplas Camadas	6
3 Conclusões	7
3.1 Considerações Finais	7
3.2 Sugestões para Pesquisas Futuras	7
A Sequências	9
Referências Bibliográficas	11
Índice Remissivo	12

Lista de Abreviaturas

CFT	Transformada contínua de Fourier (<i>Continuous Fourier Transform</i>)
DFT	Transformada discreta de Fourier (<i>Discrete Fourier Transform</i>)
EIIP	Potencial de interação elétron-íon (<i>Electron-Ion Interaction Potentials</i>)
STFT	Transformada de Fourier de tempo reduzido (<i>Short-Time Fourier Transform</i>)

Lista de Símbolos

ω	Frequência angular
ψ	Função de análise <i>wavelet</i>
Ψ	Transformada de Fourier de ψ

Lista de Figuras

2.1	Grafo de fluxo do sinal do perceptron	3
2.2	Grafo de fluxo do sinal do perceptron	4
2.3	Número máximo de iterações	6

Lista de Tabelas

A.1 Exemplo de tabela. 10

1

1.3 Contribuições

As principais contribuições deste trabalho são as seguintes:

- Item 1. Texto texto.
- Item 2. Texto texto.

1.4 Organização do Trabalho

No Capítulo [2](#), apresentamos os conceitos ... Finalmente, no Capítulo [3](#) discutimos algumas conclusões obtidas neste trabalho. Analisamos as vantagens e desvantagens do método proposto ...

As sequências testadas no trabalho estão disponíveis no Apêndice [A](#).

Capítulo 2

Conceitos

2.1 Perceptrons de Camada Única

O perceptron é construído em torno de um neurônio não-linear, isto é, o modelo de *MacCulloch-Pitts* de um neurônio. Este modelo de neurônio consiste de um combinador linear seguido por um limitador abrupto (realizando a função sinal), como apresentado na Figura 2.1. O nó aditivo do modelo neuronal calcula uma combinação linear das entradas aplicadas às suas sinapses e também incorpora um bias aplicado externamente. A soma resultante, isto é, o campo local induzido, é aplicado ao limitador abrupto.

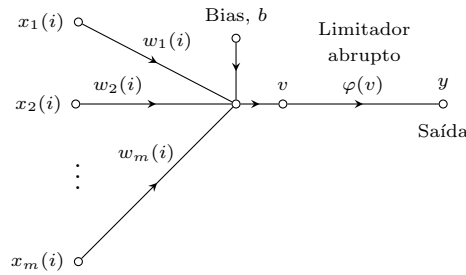


Figura 2.1: Grafo de fluxo do sinal do perceptron

2.1.1 Teorema de convergência do perceptron

Para derivar o algoritmo de aprendizagem por correção de erro para o perceptron, achamos mais conveniente trabalhar com o modelo modificado do grafo de fluxo de sinal da Figura 2.2. Neste segundo modelo, que é equivalente àquele da Figura 2.1, o bias $b(n)$ é tratado como um peso sináptico acionado por uma entrada fixa igual a +1. Podemos assim definir o vetor de entrada $(m + 1)$ -por-1

$$\mathbf{x}(n) = [+1, x_1(n), x_2(n), \dots, x_m(n)]^T$$

onde n representa o passo de iteração na aplicação do algoritmo. Correspondentemente, definimos o vetor de pesos $(m + 1)$ -por-1 como

$$\mathbf{w}(n) = [b(n), w_1(n), w_2(n), \dots, w_m(n)]^T$$

Correspondentemente, a saída do combinador linear pode ser escrita na forma compacta

$$\begin{aligned} v(n) &= \sum_{i=0}^m w_i(n) x_i(n) \\ &= \mathbf{w}^T(n) \mathbf{x}(n) \end{aligned} \tag{2.1}$$

onde $w_0(n)$ representa o bias $b(n)$. Para n fixo, a equação $\mathbf{w}^T \mathbf{x} = 0$, traçada em um espaço multidimensional (traçada para um bias determinado) com coordenadas x_1, x_2, \dots, x_m , define um hiperplano como a superfície de decisão entre duas classes diferentes de entradas.

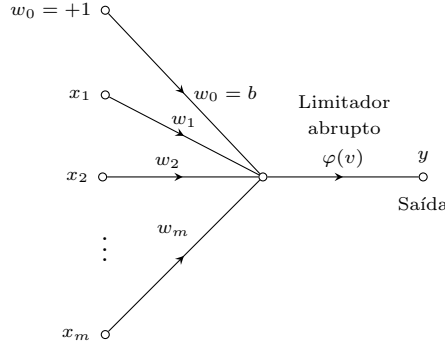


Figura 2.2: Grafo de fluxo do sinal do perceptron

Para o perceptron funcionar corretamente, as duas classes \mathcal{C}_1 e \mathcal{C}_2 devem ser *linearmente separáveis*. Por sua vez, isto significa que os padrões a serem classificados devem estar suficientemente separados entre si para assegurar que a superfície de decisão consista de um hiperplano.

Suponhamos então que as variáveis de entrada do perceptron se originem de duas classes linearmente separáveis. Seja \mathcal{X}_1 o subconjunto de vetores de treinamento $\mathbf{x}_1(1), \mathbf{x}_1(2), \dots$ que pertencem à classe \mathcal{C}_1 e seja \mathcal{X}_2 o subconjunto de vetores de treinamento $\mathbf{x}_2(1), \mathbf{x}_2(2), \dots$ que pertencem à classe \mathcal{C}_2 . A união de \mathcal{X}_1 e \mathcal{X}_2 é o conjunto de treinamento completo \mathcal{X} . Dados os conjuntos de vetores \mathcal{X}_1 e \mathcal{X}_2 para treinar o classificador, o processo de treinamento envolve o ajuste de peso \mathbf{w} de tal forma que as duas classes \mathcal{C}_1 e \mathcal{C}_2 sejam linearmente separáveis. Isto é, existe um vetor de peso \mathbf{w} para o qual podemos afirmar

$$\begin{aligned} \mathbf{w}^T \mathbf{x} &> 0 \text{ para todo vetor de entrada } \mathbf{x} \text{ pertencente à classe } \mathcal{C}_1 \\ \mathbf{w}^T \mathbf{x} &\leq 0 \text{ para todo vetor de entrada } \mathbf{x} \text{ pertencente à classe } \mathcal{C}_2 \end{aligned} \quad (2.2)$$

Na segunda linha da Equação 2.2, escolhemos arbitrariamente que o vetor de entrada \mathbf{x} pertence à classe \mathcal{C}_2 se $\mathbf{w}^T \mathbf{x} = 0$. Dados os subconjuntos de vetores de treinamento \mathcal{X}_1 e \mathcal{X}_2 , o problema de treinamento para o perceptron elementar é, então, encontrar um vetor de peso \mathbf{w} tal que as duas desigualdades da Equação 2.2 sejam satisfeitas.

O algoritmo para adaptar o vetor de peso do perceptron elementar pode ser formulado como segue:

1. Se o n -ésimo membro do conjunto de treinamento, $\mathbf{x}(n)$, é corretamente classificado pelo vetor de peso $\mathbf{w}(n)$ calculado na n -ésima iteração do algoritmo, então o vetor de peso do perceptron não é corrigido de acordo com a regra:

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) & \text{se } \mathbf{w}^T(n)\mathbf{x}(n) > 0 \text{ e } \mathbf{x}(n) \text{ pertencente à classe } \mathcal{C}_1 \\ \mathbf{w}(n+1) &= \mathbf{w}(n) & \text{se } \mathbf{w}^T(n)\mathbf{x}(n) \leq 0 \text{ e } \mathbf{x}(n) \text{ pertencente à classe } \mathcal{C}_2 \end{aligned} \quad (2.3)$$

2. Caso contrário, o vetor de peso do perceptron é atualizado de acordo com a regra

$$\begin{aligned} \mathbf{w}(n+1) &= \mathbf{w}(n) - \eta(n)\mathbf{x}(n) & \text{se } \mathbf{w}^T(n)\mathbf{x}(n) > 0 \text{ e } \mathbf{x}(n) \text{ pertencente à classe } \mathcal{C}_2 \\ \mathbf{w}(n+1) &= \mathbf{w}(n) + \eta(n)\mathbf{x}(n) & \text{se } \mathbf{w}^T(n)\mathbf{x}(n) \leq 0 \text{ e } \mathbf{x}(n) \text{ pertencente à classe } \mathcal{C}_1 \end{aligned} \quad (2.4)$$

onde o *parâmetro da taxa de aprendizagem* $\eta(n)$ controla o ajuste aplicado ao vetor de peso na iteração n .

Se $\eta(n) = \eta > 0$, onde η é uma constante independente do número da iteração n , temos uma *regra de adaptação com incremento fixo* para o perceptron.

No que segue, primeiro provamos a convergência de uma regra de adaptação com incremento fixo para a qual $\eta = 1$. Claramente, o valor de η não é importante, desde que seja positivo. Um valor de $\eta \neq 1$ meramente escala os vetores de padrões sem afetar a sua separabilidade.

A prova é apresentada para a condição inicial $\mathbf{w}(0) = \mathbf{0}$. Suponhamos que $\mathbf{w}^T(n)\mathbf{x}(n) < 0$ para $n = 1, 2, \dots$, e que o vetor de entrada $\mathbf{x}(n)$ pertença ao subconjunto \mathcal{X}_1 . Isto é, o perceptron classifica incorretamente os vetores $\mathbf{x}(1), \mathbf{x}(2), \dots$, já que a segunda condição da Equação 2.2 é violada. Então, com a constante $\eta(n) = 1$, podemos a segunda linha da Equação 2.4 para escrever

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \mathbf{x}(n) \quad \text{para } \mathbf{x}(n) \text{ pertencente à classe } \mathcal{C}_1. \quad (2.5)$$

Dada a condição inicial $\mathbf{w}(0) = \mathbf{0}$, podemos resolver iterativamente esta equação para $\mathbf{w}(n+1)$ obtendo o resultado

$$\mathbf{w}(n+1) = \mathbf{x}(1) + \mathbf{x}(2) + \dots + \mathbf{x}(n) \quad (2.6)$$

Como as classes \mathcal{C}_1 e \mathcal{C}_2 são assumidas como sendo linearmente separáveis, existe uma solução \mathbf{w}_0 para a qual $\mathbf{w}^T(n)\mathbf{x}(n) > 0$ para os vetores $\mathbf{x}(1), \dots, \mathbf{x}(n)$ pertencentes ao subconjunto \mathcal{X}_1 . Para uma solução fixa \mathbf{w}_0 , podemos então definir um número positivo α como

$$\alpha = \min_{\mathbf{x}(n) \in \mathcal{X}_1} \mathbf{w}_0^T \mathbf{x}(n) \quad (2.7)$$

Assim, multiplicando ambos os lados da Equação 2.6 pelo vetor linha \mathbf{w}_0^T , obtemos

$$\mathbf{w}_0^T \mathbf{w}(n+1) = \mathbf{w}_0^T \mathbf{x}(1) + \mathbf{w}_0^T \mathbf{x}(2) + \dots + \mathbf{w}_0^T \mathbf{x}(n)$$

Consequentemente, com base na definição dada na Equação 2.7, temos

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

Da *desigualdade de Cauchy-Schwarz* segue que

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

ou de forma equivalente,

$$\|\mathbf{w}(n+1)\|^2 \geq \frac{n^2 \alpha^2}{\|\mathbf{w}_0\|^2} \quad (2.8)$$

A seguir, seguimos com um outro caminho de desenvolvimento. Em particular, reescrevemos a Equação 2.5 na forma

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{x}(k) \quad \text{para } k = 1, \dots, n \text{ e } \mathbf{x}(k) \in \mathcal{X}_1 \quad (2.9)$$

Calculando a norma euclidiana quadrática de ambos os lados da Equação 2.9, obtemos

$$\|\mathbf{w}(k+1)\|^2 = \|\mathbf{w}(k)\|^2 + \|\mathbf{x}(k)\|^2 + 2\mathbf{w}^T(k)\mathbf{x}(k) \quad (2.10)$$

Mas, sob a suposição que o perceptron classifica incorretamente um vetor de entrada $\mathbf{x}(k)$ pertencente ao subconjunto \mathcal{X}_1 , temos $\mathbf{w}^T(k)\mathbf{x}(k) < 0$. Consequentemente, deduzimos da Equação 2.10 que

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

Somando estas desigualdades para $k = 1, \dots, n$ e invocando a condição inicial assumida $\mathbf{w}(0) = \mathbf{0}$, obtemos a desigualdade:

$$\begin{aligned} \|\mathbf{w}(k+1)\|^2 &\leq \sum_{k=1}^n \|\mathbf{x}(k)\|^2 \\ &\leq n\beta \end{aligned} \quad (2.11)$$

onde β é um número positivo definido por

$$\beta = \max_{\mathbf{x}(k) \in \mathcal{X}_1} \|\mathbf{x}(k)\|^2 \quad (2.12)$$

A Equação 2.11 afirma que a norma euclidiana quadrática do vetor de peso $\mathbf{w}(n+1)$ cresce no máximo linearmente como o número de iterações n . O segundo resultado da Equação 2.11 está claramente em conflito com o resultado anterior da Equação 2.8 para valores suficientemente grandes de n . De fato, podemos afirmar que n não pode ser maior que um valor n_{\max} para o qual as Equações 2.8 e 2.11 são ambas satisfeitas com um sinal de igualdade. Isto é, dada uma solução \mathbf{w}_0 temos

$$n_{\max} = \frac{\beta \|\mathbf{w}_0\|^2}{\alpha^2} \quad (2.13)$$

Provamos assim que para $\eta(n) = 1$ para todo n , e $\mathbf{w}(0) = \mathbf{0}$, e desde que exista um vetor solução \mathbf{w}_0 , a regra para adaptar os pesos sinápticos do perceptron deve terminar após no máximo n_{\max} iterações. Note também que das Equações 2.7, 2.12 e 2.13 que não existe uma solução única para \mathbf{w}_0 ou n_{\max} .

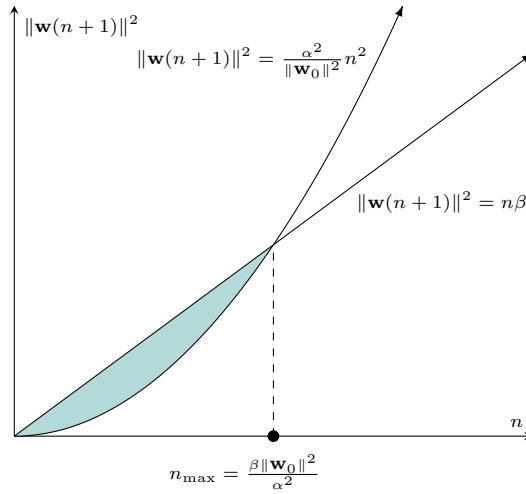


Figura 2.3: Número máximo de iterações

Procedure 1 Algoritmo de convergência do perceptron

- 1: $\mathbf{X}_1 \leftarrow$ Entradas pertencentes à classe \mathcal{X}_1
 - 2: $\mathbf{X}_2 \leftarrow$ Entradas pertencentes à classe \mathcal{X}_2
 - 3: Inicialize \mathbf{w} com zeros
 - 4: **while** !convergence **do**
 - 5: Escolha aleatoriamente $\mathbf{x} \in \mathbf{X}_1 \cup \mathbf{X}_2$
 - 6: **if** $\mathbf{x} \in \mathbf{X}_1$ e $\mathbf{w} \cdot \mathbf{x} < 0$ **then**
 - 7: $\mathbf{w} = \mathbf{w} + \mathbf{x}$
 - 8: **end if**
 - 9: **if** $\mathbf{x} \in \mathbf{X}_2$ e $\mathbf{w} \cdot \mathbf{x} \geq 0$ **then**
 - 10: $\mathbf{w} = \mathbf{w} - \mathbf{x}$
 - 11: **end if**
 - 12: **end while**
-

2.2 Perceptrons de Múltiplas Camadas

Conclusões

3.1 Considerações Finais

3.2 Sugestões para Pesquisas Futuras

Finalmente, leia o trabalho de Uri Alon [Alo09] no qual apresenta-se uma reflexão sobre a utilização da Lei de Pareto para tentar definir/escolher problemas para as diferentes fases da vida acadêmica. A direção dos novos passos para a continuidade da vida acadêmica deveriam ser discutidos com seu orientador.

7

Apêndice A

Sequências

[illegible]

<i>Limiar</i>	MGWT			AMI			<i>Spectrum</i> de Fourier			Características espectrais		
	<i>Sn</i>	<i>Sp</i>	<i>AC</i>	<i>Sn</i>	<i>Sp</i>	<i>AC</i>	<i>Sn</i>	<i>Sp</i>	<i>AC</i>	<i>Sn</i>	<i>Sp</i>	<i>AC</i>
1	1.00	0.16	0.08	1.00	0.16	0.08	1.00	0.16	0.08	1.00	0.16	0.08
2	1.00	0.16	0.09	1.00	0.16	0.09	1.00	0.16	0.09	1.00	0.16	0.09
2	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10
4	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10	1.00	0.16	0.10
5	1.00	0.16	0.11	1.00	0.16	0.11	1.00	0.16	0.11	1.00	0.16	0.11
6	1.00	0.16	0.12	1.00	0.16	0.12	1.00	0.16	0.12	1.00	0.16	0.12
7	1.00	0.17	0.12	1.00	0.17	0.12	1.00	0.17	0.12	1.00	0.17	0.13
8	1.00	0.17	0.13	1.00	0.17	0.13	1.00	0.17	0.13	1.00	0.17	0.13
9	1.00	0.17	0.14	1.00	0.17	0.14	1.00	0.17	0.14	1.00	0.17	0.14
10	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15
11	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15	1.00	0.17	0.15
12	1.00	0.18	0.16	1.00	0.18	0.16	1.00	0.18	0.16	1.00	0.18	0.16
13	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17
14	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17	1.00	0.18	0.17
15	1.00	0.18	0.18	1.00	0.18	0.18	1.00	0.18	0.18	1.00	0.18	0.18
16	1.00	0.18	0.19	1.00	0.18	0.19	1.00	0.18	0.19	1.00	0.18	0.19
17	1.00	0.19	0.19	1.00	0.19	0.19	1.00	0.19	0.19	1.00	0.19	0.19
17	1.00	0.19	0.20	1.00	0.19	0.20	1.00	0.19	0.20	1.00	0.19	0.20
19	1.00	0.19	0.21	1.00	0.19	0.21	1.00	0.19	0.21	1.00	0.19	0.21
20	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22	1.00	0.19	0.22

Tabela A.1: *Exemplo de tabela.*

Referências Bibliográficas

- [Alo09] Uri Alon. How To Choose a Good Scientific Problem. *Molecular Cell*, 35(6):726–728, Setembro 2009. [7](#)
- [Eco09] Umberto Eco. *Como se Faz uma Tese*. Perspectiva, 22^o edição, 2009. Tradução Gilson Cesar Cardoso de Souza. [1](#)
- [Hig98] Nicholas J. Higham. *Handbook of Writing for the Mathematical Sciences*. SIAM: Society for Industrial and Applied Mathematics, segunda edição, Agosto 1998. [1](#)
- [KLR96] Donald E. Knuth, Tracy Larrabee e Paul M. Roberts. *Mathematical Writing*. The Mathematical Association of America, Setembro 1996. [1](#)
- [Tuf01] Edward Tufte. *The Visual Display of Quantitative Information*. Graphics Pr, 2nd edição, Maio 2001. [1](#)
- [Waz09] Raul S. Wazlawick. *Metodologia de Pesquisa em Ciencia da Computação*. Campus, primeira edição, 2009. [1](#)
- [Zob04] Justin Zobel. *Writing for Computer Science: The art of effective communication*. Springer, segunda edição, 2004. [1](#)

Índice Remissivo

DFT, *veja* transformada discreta de Fourier

DSP, *veja* processamento digital de sinais

Fourier

transformada, *veja* transformada de Fourier

genoma

projetos, [1](#)

STFT, *veja* transformada de Fourier de tempo

reduzido

TBP, *veja* periodicidade região codificante

área do trabalho

fundamentos, [3](#)