

UNIVERSIDADE DE SÃO PAULO
INSTITUTO DE FÍSICA

Capacidade Máxima do Percéptron num Modelo
para Discriminar Odores

Juan Ernesto Palo Tejada

Dissertação apresentada ao Instituto de
Física da Universidade de São Paulo
par obtenção do título de Mestre em
Ciências

Orientador : Prof. Dr. Nestor Felipe Caticha Alfonso

Banca Examinadora:

Prof. Dr.

Prof. Dr.

Prof. Dr. Nestor Felipe Caticha Alfonso

SÃO PAULO

2000

À Martha

Agradecimentos

Ao Nestor Caticha pela orientação.

A FAPESP pelo apoio financeiro.

Abstract

Olfaction is based on an array of a few hundred different types of sensory neurons, each expressing one or a few types of molecular receptors. Together, this neuronal ensemble is capable of recognizing all odorants and measuring their concentrations. Recently was proposed a probabilistic receptor affinity distribution model, which describes the statistical properties of receptor ensembles such as the one expressed in the olfactory system. This model introduces an affinity distribution $\rho(q)$ for the values of binding constant q between an arbitrary ligand (odorant) and a collection of receptors. Since a given sensory neuron is likely to possess receptors of a single kind, the magnitudes of responses of different sensory neurons to a given odorant μ is determined by the affinities of their receptors, and hence can be inferred from $\rho(q)$.

The first step in information processing in the olfactory pathway is known to take place in the olfactory bulb. The axons of the sensory neurons project to this target and form synapses on secondary neurons (mitral and tufted-cells). Inter-neurons of two major types (periglomerular and granule cells) are believed to play role in computing the pattern transmitted from the olfactory bulb to higher brain centers. In this thesis we address some of the potential computational characteristics of the olfactory bulb.

The model we introduce is also interesting from a mathematical point of view since it generalizes the problem of Linear Separability (LS) of points in N dimensional space to that of separability of strings or curves which are parametrized continuously by the odorant concentration. We restrict ourselves to a numerical analysis of the capacity of his neural network, related to the number of odorants the can be distinguished. The model, based on a single layer perceptron, is introduced and discussed in detail in chapter 2 in chapter 3 we present the method to determine numerically the capacity. This is based on a learning algorithm by Nabutovsky and Domany which is capable of, not only finding the separation plane-when it exists- but also of signaling the fact that a sample of examples is not LS.

Although this problem lacks a characteristic length scale, it is possible to carry out a finite size scaling (FSS) analysis of the data. The main results are presented as curves of capacity as a function of odorant concentration in the thermodynamic limit, where we let the number of receptor units N to infinity via FSS..

Resumo

O sistema olfativo está baseado num arranjo de algumas centenas de neurônios sensores de diferentes classes. Cada neurônio representa um ou muitos tipos de receptores moleculares. Juntos, esses neurônios podem reconhecer todos os odorantes e medir suas concentrações. Recentemente, foi proposto um modelo probabilístico de distribuição de afinidades em receptores, descrevendo as propriedades estatísticas do conjunto destes que caracterizam o sistema olfativo. Esse modelo introduz uma distribuição de afinidades $\rho(q)$, que relaciona uma constante q a um odorante arbitrário e a uma coleção de receptores. Desde que é provável que um neurônio sensor tenha um só tipo de receptores, a magnitude de saída do neurônio sensor é diferente para cada odor μ , sendo determinada pela afinidade de seus receptores e pode-se inferir de $\rho(q)$.

O primeiro passo no processamento da informação, no processo de percepção de odores, acontece no vulvo olfativo. Os axônios dos neurônios sensoriais são projetados formando as sinapses dos neurônios secundários ("mitral" e "tufted cells"). Inter-neurônios de 2 tipos principais (periglomerular e granule cells), acredita-se, têm a função de transmitir os dados do vulvo olfativo às regiões mais altas do cérebro. Nesta dissertação, estudaremos algumas das características computacionais do vulvo olfativo, por meio de um modelo bastante simples.

O modelo introduzido nesta dissertação também é importante do ponto de vista matemático, desde que generaliza o problema de separação linear (LS) de pontos no espaço de N dimensões ao problema de separar curvas que são continuamente parametrizadas pela concentração de odorantes. Restringiremo-nos à análise numérica da capacidade desta Rede Neural, que diz do número de odorantes que podem ser separados. O modelo baseado no percéptron sim camada escondida é introduzido no capítulo 2. No capítulo 3, apresentamos o método para determinar numericamente a capacidade. Tal método está baseado no algoritmo de Nabutovsky e Domany, que é capaz não só de descobrir se um plano de separação existe, como também é capaz de determinar se um conjunto de exemplos é linearmente não-separável.

Dado que este problema não tem uma escala de comprimento característica, é possível fazer um análise de dados "finite size scaling (FSS)" Os resultados principais são apresentados em forma de curvas de capacidade em função da concentração do odorante no limite termodinâmico, onde levamos o número de receptores N ao infinito via FSS.

Sumário

1	Introdução	10
1.1	Histórico	10
1.2	Percéptron	12
1.2.1	Problemas Linearmente separáveis	14
1.2.2	Aprendizado Supervisionado	16
1.3	Dinâmica de Aprendizagem	17
1.3.1	Dinâmica offline	17
1.3.2	Dinâmica Online	17
1.4	O Sistema Biológico de Percepção de Odores	18
1.4.1	Descrição	18
1.4.2	Mecanismo	19
1.5	Justificação	20
1.6	Organização da tese	21
2	O Conjunto de treinamento	23
2.1	O Modelo	23
2.2	O Conjunto de treinamento	26
2.2.1	Construindo os vetores de afinidade	26
2.2.2	Construindo o conjunto de treinamento	28
2.2.3	Análise do Conjunto de Treinamento e Mudança da representação	28
3	O algoritmo de treinamento	33
3.1	Introdução	33
3.2	O Algoritmo	34

3.3	Análises do algoritmo	35
3.3.1	Prova de $\frac{\vec{w} \cdot \vec{w}^*}{h_{min}^*} \geq d$	36
3.3.2	Prova de $h_{min}^* \geq \frac{1}{d_c}$	37
3.4	Teorema de Convergência	39
3.5	O crescimento exponencial de d	41
3.6	Simulação	43
3.6.1	Caso Linearmente Separável	46
3.6.2	Caso Linearmente Não separável	48
3.7	Conclusão	50
4	Simulação	52
4.1	Escolha dos parâmetros de treinamento	54
4.1.1	O número de concentrações por odorante	55
4.2	Construindo as curvas $P(h,p/N)$ vs. p/N	57
4.3	Capacidade crítica no problema de odorantes α_c	60
4.3.1	Análise FFS	60
4.3.2	Simulações	61
5	Conclusões	72
5.1	Interpretação geométrica	72
5.2	Conclusões	76

Capítulo 1

Introdução

1.1 Histórico

As primeiras tentativas de modelar redes neurais (RN) surgiram na década do 40, com o trabalho seminal de McCulloch e Pitts[1]. Este trabalho é importante por muitas razões, além do fato consistente de que foi o primeiro em tratar o cérebro como um sistema computacional. A teoria de McCulloch e Pitts está basicamente fundamentada em 5 suposições:

1. A atividade de um neurônio é um processo booleano.
2. É preciso que um número fixo de sinápses sejam excitadas dentro de um período de tempo para que um neurônio seja excitado.
3. O único retardo significativo dentro do sistema nervoso é o retardo sináptico.
4. A atividade de quaisquer sinápses de inibição impede a excitação do neurônio.
5. A arquitetura da rede não é modificada com o tempo.

A teoria de McCulloch-Pitts não é um modelo preciso da atividade cerebral, entretanto a importância desse trabalho está em ele ser um dos primeiros que levaram aos modelos de RN. Em 1949 Hebb[2] em seu livro *“Organization of Behaviour”* sugeria uma resposta à pergunta: como aprendemos? Dado que as conexões entre neurônios são feitas através da sinápses, para Hebb, era razoável supor que quaisquer

modificação que pudesse acontecer durante a aprendizagem deveria acontecer nelas. Hebb defendia a teoria de que assim a área da união sináptica aumentava. Teorias mais recentes afirmam que o responsável é um incremento na velocidade com que se libera o neurotransmissor em a célula sináptica. Do mesmo modo que o modelo de McCulloch e Pitts, o modelo de Hebb não é completo. Não entanto, aparece de muitas formas nos modelos de redes neurais que existem na atualidade.

Em 1959 o psicólogo Frank Rosenblatt[3] inventou o percéptron, num intento de tratar algumas das propriedades fundamentais dos sistemas inteligentes, Rosenblatt poss demonstrar que o percéptron era capaz de classificar padrões corretamente, no que ele chamava de entorno diferenciado, o percéptron de rosenblatt também era capaz de responder de maneira congruente frente à tramas aleatórias.

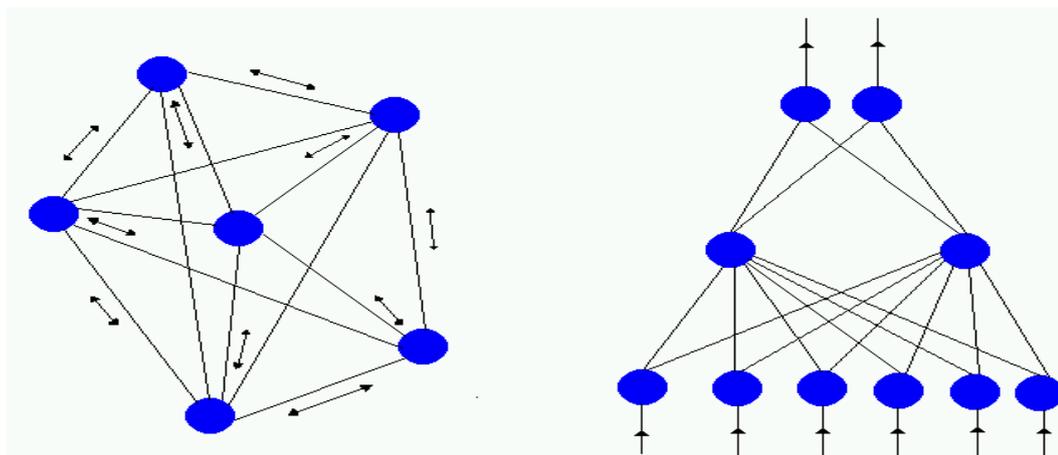
Em 1969 o livro do Marvin Minsky e Seymour Paper[4], ambos do MIT, chamado “Percéptron: An Introduction to Computational Geometry”, mostrava uma detalhada análise do percéptron em termos de suas capacidades e limitações. O percéptron booleano, a versão mas conhecida desta máquina, só implementa funções linearmente separáveis, como muitas funções de importância prática não o são, o iteres por RN diminuiu até quase desaparecer, no final da década do 60. Sabia-se que redes com camadas intermediárias podiam implementar funções mais complicadas, mas neste caso o problema era a inexistência de um algoritmo para treinar a rede.

Na década do 80 estes problemas foram superados quando surgiu o algoritmo de retro-propagação de Rumelhart, Hinton e Willians[5] “blackpropagation”, aplicável a RNCs (redes com arquitetura de acoplamentos unidirecionais e organizadas em camadas “feedforward”) com funções de transferência deriváveis. Nesta mesma década Hopfield[6] relacionou as RNs conhecidas como redes de atratores aos vidros de spin. A proposta de Hopfield, que generaliza o modelo de Little[7], envolvia um tipo de modelo composto por neurônios com dois estados $S_i \in \{+1, -1\}$ interagindo de maneira simétrica através de acoplamentos sinápticos w_{ij} modulados segundo a proposta de Hebb.

As diversas possibilidades de aplicação prática das RNCs, como o aperfeiçoamento dos métodos de Mecânica Estatística no estudo das redes, propiciaram um grande crescimento desta área.

No final da década do 80 o trabalho de Gardner e Derrida[8] teve grande re-

percussão. A idéia de Gardner consistia em estudar as propriedades estatísticas do espaço dos acoplamentos sinápticos w_{ij} ao invés do espaço das configurações de atividade S_j . Isto possibilitou o estudo de diversas situações novas, entre estas o problema de generalização no cenário professor-aluno.



Redes de atratores

Redes Feedforward

Figura 1.1

1.2 Percéptron

O percéptron é uma RNCs de uma camada com N neurônios de entrada conectados através de sinápses w_i , e somente uma saída O , cujo estado é dado por:

$$O = f(\vec{S}, \vec{w}), \quad (1.1)$$

O percéptron realiza um mapeamento de entrada \rightarrow saída ; neste trabalho trataremos do caso em que a entrada é um vetor $\vec{X} \in \{+\mathfrak{R}\}^N$ e a saída é uma variável booleana $O = \pm 1$.

O vetor \vec{w} é um vetor $N_p - dimensional$ com $w_i \in \mathfrak{R}$, $i = 1, \dots, N_p$, \vec{w} é chamado de vetor sináptico ou de conexões, e parametriza a função implementada pelo

percéptron. A função de transferência $f(x)$ usada neste trabalho é $f(x) = \text{sign}(x)$. No entanto, várias funções de transferência são usadas nos estudos e aplicações de RNs, por exemplo:

- função de transferência linear $f(x) = x$
- função de transferência analógica $f(x) = \tan(x)$
- função de transferência booleana $f(x) = \text{sign}(x)$
- função de transferência analógica $f(x) = (1 - e^{-x})^{-1}$

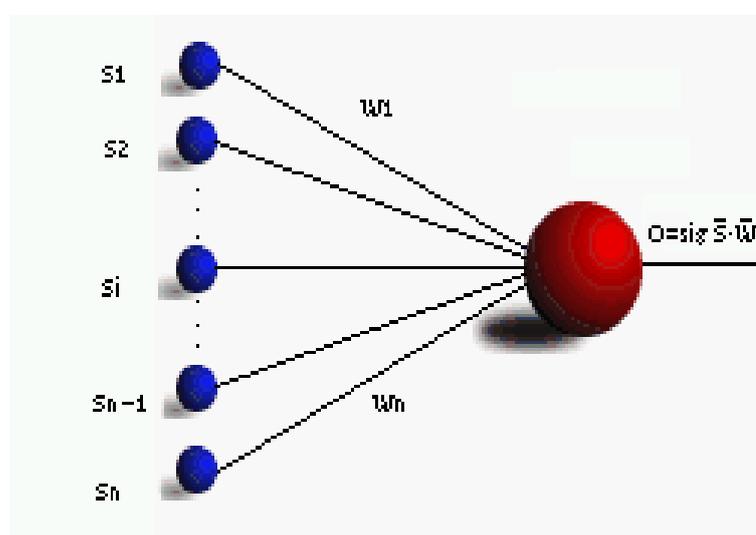


Figura 1.2: Percéptron booleano de entradas S_i e saída escalar $O = \text{sign}(\vec{S} \cdot \vec{w})$.

Uma vez que fixamos a função de transferência, a função implementada pelo percéptron (que para $f(x) = \text{sign}(x)$ é chamado de percéptron booleano) é completamente definida por \vec{w} . Cada componente w_i do vetor sináptico pode ser compreendida como uma variável que multiplica a componente S_i do vetor de entrada, na composição do chamado campo pós-sináptico $h = \vec{S} \cdot \vec{w}$ que determina a saída da rede, figura 1.2. Dado um particular vetor ou padrão de entrada \vec{S}^μ , a saída correspondente O^μ do percéptron booleano pode ser compreendida como uma classificação deste

padrão entre duas categorias distintas que ficam definidas a partir do vetor sináptico \vec{w} . Este define um hiperplano de $N_p - 1$ dimensões que divide o espaço de N_p dimensões em duas regiões, uma das quais contém o vetor \vec{w} .

Dado um par (\vec{S}, O) com $O \in \{-1, +1\}$, sempre existe um vetor \vec{w} tal que $O = O(\vec{S})$. Dado um conjunto de P pares aleatórios $\{(\vec{S}^\mu, O^\mu)\}, \mu = 1, \dots, P$, existe algum \vec{w} para o qual $O^\mu = O(\vec{S}^\mu)$, para todo $1 < \mu < P$? Esta pergunta diz respeito à capacidade do percéptron de reproduzir pares entrada-saída, através da obtenção do vetor \vec{w} apropriado (caso este exista). De maneira geral, o problema de obtenção de \vec{w} a partir de um conjunto de pares entrada-saída é chamado de aprendizagem.

1.2.1 Problemas Linearmente separáveis

A estrutura de um percéptron é completamente determinada por o conjunto de sinápses $w_i, i = 1, \dots, N$, onde N é o número de entradas da RN, dada uma função Booleana definida por o conjunto $\ell = \{\xi_i^\mu, O^\mu\}$, com $\xi_i^\mu \in \{-1, +1\}$ e 2 possíveis estados de saída $O^\mu = \text{sign}(\sum_{i=0}^N w_i \xi_i^\mu)$. A função Booleana que pode ser implementada por um percéptron simples de N entradas e uma saída de um subconjunto de todas as possíveis funções Booleanas são chamadas funções linearmente separáveis.

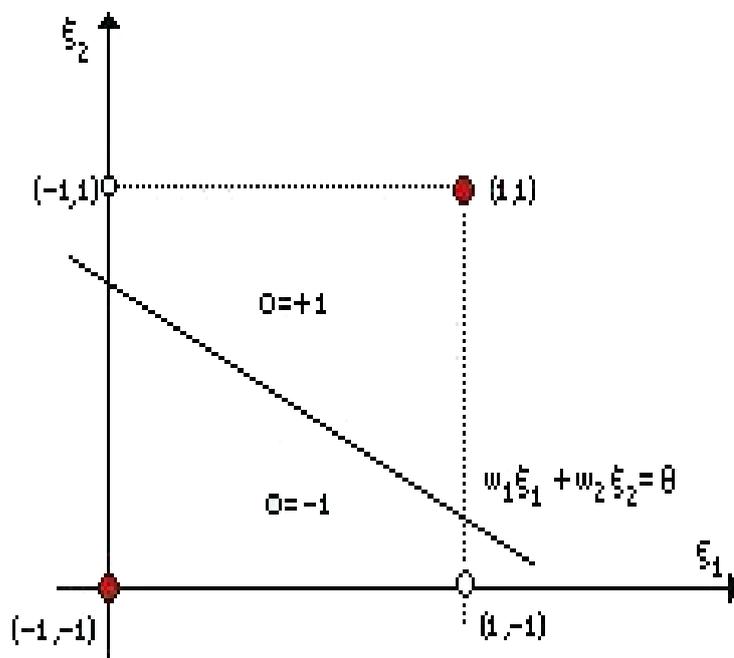


Figura 1.3 A função XOR é linearmente não separável. Não é possível encontrar uma linha reta que separe os pontos pretos dos brancos.

A função XOR por exemplo não pode ser implementada por um percéptron com $N = 2$, o que diz que a função XOR é linearmente não separável. Isto é devido ao campo do percéptron é dado por:

$$h^\mu = w_1 \xi_1^\mu + w_2 \xi_2^\mu + \theta,$$

esta é a equação de uma linha reta no plano (ξ_1, ξ_2) de parâmetros w_1 , w_2 e θ . O problema é encontrar um conjunto de parâmetros que separem os pontos $(+1, +1)$ e $(-1, -1)$ dos pontos $(+1, -1)$ e $(-1, +1)$ (ver figura 1.3). Não existe um conjunto de parâmetros que possa separar estes pontos, portanto a função XOR não pode ser implementada no percéptron.

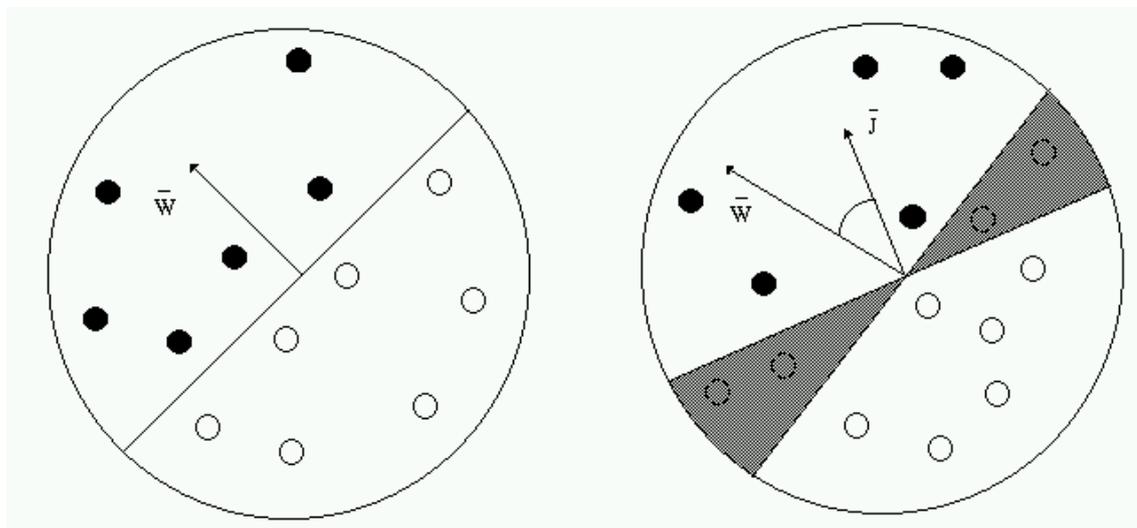


Figura 1.4 O percéptron de saída binária realiza uma classificação linearmente separável. esquerda: todas as entradas a cima do hiperplano definidas por $\vec{w} \cdot \vec{\xi} = 0$ são mapeadas por $+1(-1)$; $+1$ é representado por \bullet , -1 por \circ . Direita: O percéptron com pesos \vec{w} aprende a mapear um problema linearmente separável representado por o professor \vec{J} .

1.2.2 Aprendizado Supervisionado

Dado uma rede professor e uma outra rede aluno, chamamos de aprendizado supervisionado o processo através do qual o aluno tenta implementar a classificação do professor a partir de exemplos gerados por este. o objetivo deste processo é que a rede aluno de alguma maneira se aproxime da rede professor, fornecendo então o máximo de respostas iguais às do professor. Uma formulação mais precisa desta proximidade envolve grandezas que são chamadas de medidas de desempenho, como a capacidade de generalização e memorização. Definimos o conjunto de aprendizado ℓ como sendo o conjunto dos P exemplos gerados pelo professor que forem usados pelo aluno durante o aprendizado, $\ell = \{(\vec{S}^\mu, 0^\mu)\}$, $\mu = 1, \dots, P$. Duas grandezas podem ser definidas:

- **O erro de generalização** consiste na probabilidade de que a rede aluno classifique erroneamente um exemplo aleatório, totalmente decorrelacionado

dos exemplos do conjunto de treinamento.

- **O erro de memorização** consiste na probabilidade de que a rede aluno classifique erroneamente um exemplo totalmente correlacionado com os exemplos do conjunto de treinamento.

1.3 Dinâmica de Aprendizagem

1.3.1 Dinâmica offline

Nesse processo, as sinápses evoluem segundo uma dinâmica caracterizada por uma energia, que deve ser minimizada globalmente no processo de aprendizagem. Essa energia depende do conjunto total de exemplos (os exemplos são usados de forma paralela). Tal problema pode ser tratado como um sistema em equilíbrio termodinâmico, através dos métodos da Mecânica Estatística. A dinâmica referida é caracterizada basicamente por:

- A energia de treinamento que define a dinâmica envolve contribuições dos N exemplos.
- Toda a informação dos N exemplos é usada ao longo da dinâmica, até que o mínimo da energia livre tenha sido atingido.

1.3.2 Dinâmica Online

Na aprendizagem online, cada exemplo é sorteado do conjunto de treinamento e apresentado na rede. Isto induz uma mudança nas sinápses do aluno uma única vez. A dinâmica online apresenta um maior desempenho em tempo computacional (online é mais barato computacionalmente); entretanto, trata-se de um sistema fora do equilíbrio. Essa dinâmica é caracterizada basicamente por:

- A dinâmica de aprendizagem da rede aluno se baseia, em cada instante, na informação de um único exemplo.
- A informação de um dado exemplo é usada em um único passo da dinâmica, após o que um novo exemplo é usado para o próximo passo.

1.4 O Sistema Biológico de Percepção de Odores

No que segue, pretende-se expor de maneira muito resumida o mecanismo de percepção de odores em vertebrados, e identificar elementos comuns nos sistemas biológicos de percepção de odores e o modelo que estudaremos com detalhe no capítulo 2¹.

O Sistema de percepção de Odores (SPO), que sente e processa odores, é uma das mais velhas e mais importantes partes do cérebro. Para muitos animais este é o mais importante modo de comunicação e tem influência em muitas funções importantes, incluindo a reprodução.

1.4.1 Descrição

Recentemente foi identificada uma larga família de genes[9], que se acredita estão associados a receptores de odores². A informação em padrões de excitação destes receptores é organizada em padrões que o cérebro pode interpretar como diferentes odores.

O SPO contém células sensoriais especializadas, os neurônios, que têm fibras chamadas de “cílios” e uma única fibra nervosa chamada de axônio. Os cílios contêm receptores específicos para cada molécula de odor. Diferentes neurônios de odor respondem a diferentes odores, e odores específicos estimulam partes específicas no SPO.

Descobriram-se, até agora, 500 a 1000 tipos diferentes de receptores. Esses receptores são programados para reconhecer odores específicos, por uma das mais longas famílias de genes descoberta. Cada receptor tem codificada uma informação precisa para reagir a um odor específico. Esta informação é codificada mediante genes associadas a receptores.

¹É importante notar que a descrição que fazemos dos sistemas biológicos de percepção é uma visão simplificada. A neurobiologia é uma área muito mais complexa que a apresentada neste trabalho.

²De algum modo receptores de odores são codificados nesses genes

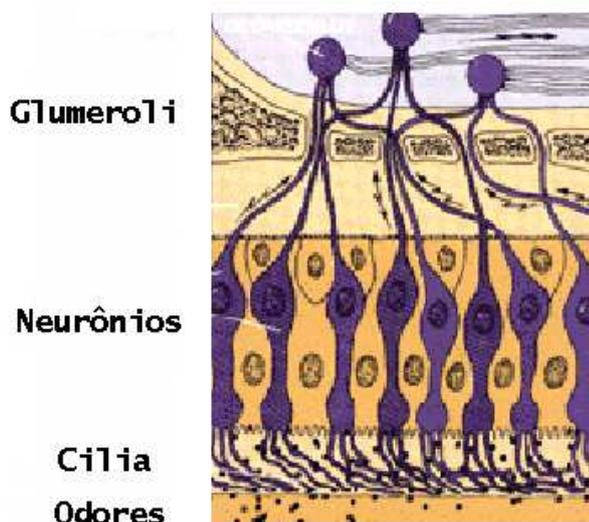


Figura 1.4: Esquema do SPO. Moléculas de odor são ligadas nos cílios do neurônio transdutor. Neurônios com receptores específicos são distribuídas aleatoriamente dentro de zonas na membrana sensorial. Sinais de neurônios com os mesmos receptores convergem a estruturas chamadas de glomeruli no bulbo olfatório.

1.4.2 Mecanismo

A membrana sensorial nos vertebrados está dividida em zonas, cada zona contendo tipos de receptores específicos. Associados a esses receptores, cada zona apresenta o mesmo tipo de genes; os genes dentro de cada zona são distribuídos aleatoriamente. Fibras sensoriais de neurônios com o mesmo tipo de receptor convergem a uma zona particular do glumeroli.³

Moléculas individuais de odor estimulam muitos tipos de receptores cada um dos quais responde por uma parte de sua estrutura molecular. Técnicas de mapeamento do cérebro mostram que partes diferentes do glumeroli são ativados por odores diferentes, formando um mapa sensorial que codifica, no glumeroli, o que o cérebro pode reconhecer como um único odor.

³Glumeroli : Estrutura especializada na qual neurônios receptores de odor se conectam com outros tipos de neurônios.

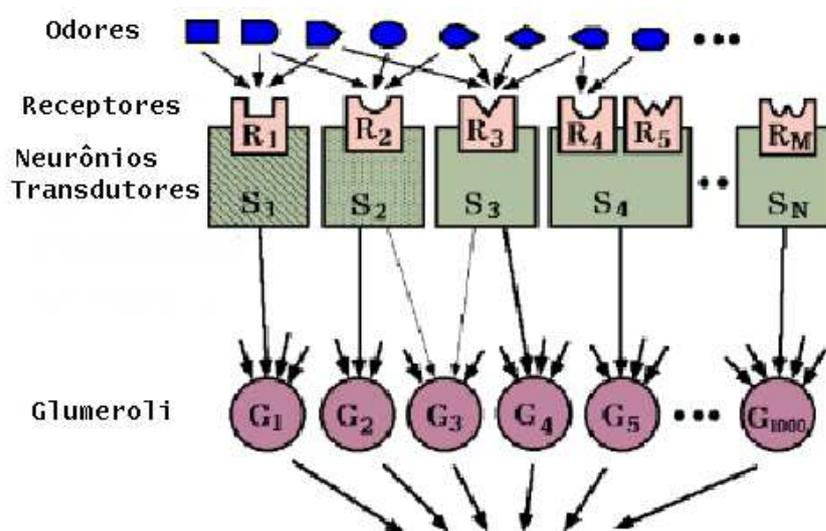


Figura 1.5: Esquema idealizado SPO. Cada célula sensora (S_i) parece ter só um tipo de receptores (R_i). Cada tipo de receptor pode reconhecer muitos odores, e vice-versa. o axônio das células sensoriais são conectados às sinapses de neurônios especiais no cérebro, chamados de glomeruli (G_i)

1.5 Justificação

Esta dissertação pretende contribuir na compreensão do mecanismo computacional que permite a percepção de odores em vertebrados, mais especificamente no homem. O sistema de percepção de odores é um tópico complexo, fora do alcance de uma só dissertação, portanto não pretendemos abarcar todos os aspectos relacionados com o SPO. Centraremos nossos esforços no mecanismo de percepção, estudando um modelo simplificado baseado na distribuição de probabilidade de afinidades (em receptores) e concentrações de moléculas de odor. Nesta dissertação este modelo é estudado pela primeira vez. Ele foi proposto em colaboração com E. Domany, D.Lancet e N. Caticha. Esperamos encontrar a capacidade máxima de um Percéptron (neurônio) para separar odores; pretendemos com isto responder a pergunta:

Quantos odores podem ser apresentados a um percéptron(neurônio) de modo que este possa distinguir um odor do resto?

Dado que Neurônios biológicos têm um número enorme de sinápses, é correto dizer que tem uma certa quantidade de odores que o neurônio poderá distinguir com probabilidade 1; passada esta quantidade, o neurônio não poderá mais separar um odor do resto.

Existe muito interesse em estudar os mecanismos pelos quais somos capazes de reconhecer odores. Entender como os odores são transferidos para a membrana das células sensoriais e como estes são reconhecidos pelo cérebro é importante em áreas distintas do conhecimento como a engenharia, biologia, medicina, etc. Estudar os mecanismos de percepção de odores contribui por exemplo em:

- Entender muitas deficiências genéticas de percepção de odores.
- Como a percepção de odores afeta as emoções e o comportamento Humano
- Feromonas⁴ têm importantes funções sociais em animais, e possivelmente em humanos.
- Construir sistemas artificiais de reconhecimento de odores.
- Elaborar estratégias para ajudar pessoas com problemas de percepção de odores.

1.6 Organização da tese

O interesse central desta tese é estudar computacionalmente um modelo probabilista de odores baseado na distribuição de afinidades e concentrações. Pretendemos encontrar a máxima capacidade de um percéptron para separar odores.

No capítulo 2, apresentamos o modelo de percepção de odorantes e construímos o conjunto de treinamento da RN. Definimos também as representação do conjunto de treinamento.

No capítulo 3, escolhemos e estudamos o algoritmo apropriado para treinar a rede e testamos o algoritmo com o conjunto de treinamento dado no capítulo 2.

⁴Moléculas especiais de odor que regulam funções hormonais

No capítulo 4, escolhemos os parâmetros de treinamento, construímos as curvas de estado, e encontramos a máxima capacidade de um perceptron para separar odores.

No capítulo 5, resumimos os principais resultados e apresentamos as conclusões.

Capítulo 2

O Conjunto de treinamento

2.1 O Modelo

O objetivo deste trabalho é estudar um modelo simples de rede neural (RN) e simular as propriedades de classificação de diferentes odorantes por um sistema olfativo. Este sistema está composto por N células receptoras R_i e um sistema de classificação (neste caso, um percéptron). As saídas das células transdutoras são conectadas as entradas do percéptron, figura 2.1. Cada célula receptora é caracterizada por um conjunto de afinidades q_i^μ (o índice i indica a célula transdutora, enquanto o índice μ indica o odorante), de modo que a resposta da mesma célula transdutora será diferente para diferentes odorantes.

Chamaremos indistintamente a R_i de célula receptora, célula transdutora, ou simplesmente de transdutor.

Dado que temos N células receptoras e um valor de afinidade por odorante μ para cada uma das células receptoras, podemos construir um vetor de afinidades que chamaremos de \vec{K}^μ :

$$\vec{K}^\mu = \{q_1^\mu, q_2^\mu, \dots, q_N^\mu\}, \quad (2.1)$$

onde a componente q_i^μ é a afinidade do transdutor R_i para o odor μ que consideramos conhecido.¹

¹Afinidade é uma característica intrínseca do transdutor, que depende basicamente de sua estrutura molecular.

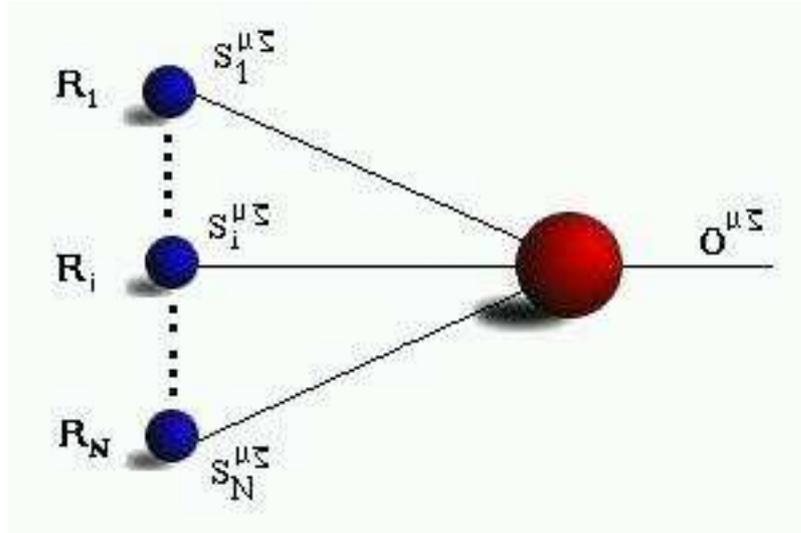


Figura 2.1: Um perceptron de N entradas e uma saída. Cada uma das entradas está conectada a um transdutor R_i . A saída do transdutor R_i é $S_i^{\mu\zeta}$, a entrada do perceptron é o vetor $\vec{S}^{\mu\zeta}$ de componentes $S_i^{\mu\zeta}$ $\{i = 1 \dots N\}$ a saída do perceptron é $\sigma^{\mu\zeta}$; ζ é o índice da concentração e μ o índice do odorante.

O sistema de classificação de odores proposto é mostrado na figura 2.1. Este sistema está composto por N células receptoras R_i de saídas $S_i^{\mu\zeta}$, as saídas de cada uma das células transdutoras e conectada a uma das entradas de um perceptron, usado como classificador. Após um processo de treinamento, o perceptron deve ter a capacidade de distinguir um odorante em particular de qualquer outro independentemente de sua concentração. A saída do perceptron é 1, se o odorante for reconhecido, e -1, se não for reconhecido.

Quando um odor μ é apresentado a um transdutor R_i , de afinidade q_i^μ para este odor, sendo H^μ a concentração de moléculas do odor μ , a resposta de R_i será:

$$S_i(q_i^\mu, H^\mu) = \frac{q_i^\mu H^\mu}{1 + q_i^\mu H^\mu}, \quad (2.2)$$

onde H^μ está entre $H_{min} \leq H \leq H_{max}$

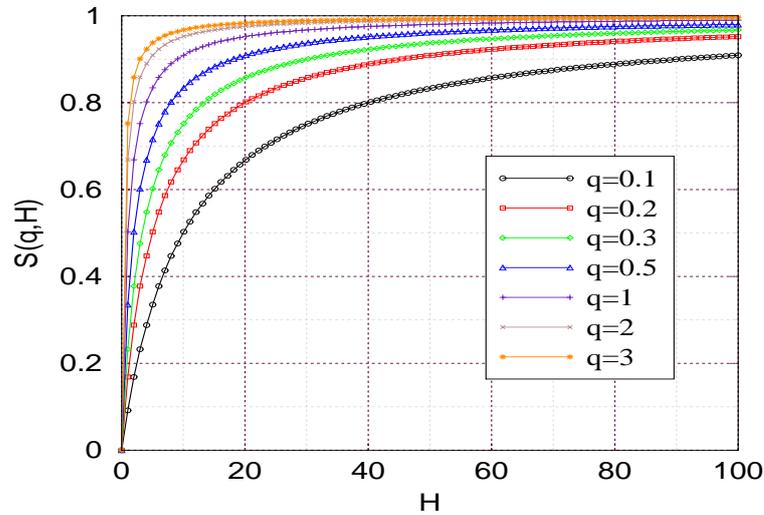


Figura 2.2: A equação (2.2) em função de q e H . Note que para baixas e altas concentrações as curvas se confundem.

Da equação (2.2) é claro que a resposta de uma célula transdutor R_i só depende da concentração do odorante e da afinidade do transdutor a este odorante. Note que a altas concentrações todas as células transdutoras apresentam a mesma resposta $S_i(q_i^\mu, H^\mu) = 1$ para qualquer odorante; em tal caso diremos que as células transdutoras estão saturadas. Células diferentes saturam a diferentes concentrações, e isto obviamente depende da afinidade de cada célula transdutora. Por outro lado a concentração mínima deve ser diferente que 0, (obviamente uma concentração 0 não tem sentido²) portanto é preciso estabelecer um valor mínimo da concentração.

De tudo isto é razoável supor que a capacidade crítica do percéptron deve ser uma função dos valores da concentração máxima H_{max} e concentração mínima H_{min} . O objetivo deste trabalho é estudar esta função.

²concentração 0 significa que não existe odorante algum

2.2 O Conjunto de treinamento

2.2.1 Construindo os vetores de afinidade

Consideraremos um conjunto de vetores \vec{K}^μ onde suas componentes serão geradas de acordo com o procedimento que segue:

Seja $\hat{\mathcal{L}}$ um conjunto de vetores que escreveremos como:

$$\hat{\mathcal{L}} = \left\{ \begin{array}{l} q_1^1, q_2^1, \dots, q_N^1 \\ q_1^2, q_2^2, \dots, q_N^2 \\ \dots\dots\dots \\ q_1^p, q_2^p, \dots, q_N^p \end{array} \right\}.$$

Onde chamaremos a $\hat{\mathcal{L}}$ conjunto de afinidades de p odores diferentes. Em forma compacta escrevemos.

$$\hat{\mathcal{L}} = \{ \vec{K}^\mu \}, \quad (2.3)$$

$$\vec{K}^\mu = \{ q_1^\mu \dots\dots\dots q_N^\mu \}.$$

Consideraremos $\hat{\mathcal{L}}$ um conjunto de vetores cujas componentes dependem das características (afinidade) dos transdutores. A afinidade q_i^μ de um transdutor para um odor μ só depende das características intrínsecas deste transdutor.

O modelo de percepção de odores baseado na distribuição de afinidades que apresentamos neste trabalho é proposto originalmente por Lancet et. al.[10].

De acordo com Lancet as afinidades de cada neurônio para cada odorante são variáveis aleatórias independentes. Elas determinam experimentalmente a forma desta distribuição que pode ser aproximada por:

$$\rho_1(q) = \frac{q}{\sigma^2} e^{-\frac{q^2}{\sigma^2}}, \quad (2.4)$$

onde cada componente q_i^μ do vetor \vec{K}^μ é uma variável aleatória independente com distribuição dada para (2.4). Note que se fazemos qualquer escolha para $\langle q \rangle$ ou

σ em (2.4), as unidades para a concentração do nosso modelo são fixadas. Para obter, numericamente, afinidades distribuídas de acordo com (2.4) introduzimos uma variável auxiliar p e procedemos da seguinte forma: Seja p uma variável aleatória independente, com distribuição $\rho_2(p)$ uniforme. De acordo com (2.4) e de $\rho_1(q)dq = \rho_2(p)dp$ segue que:

$$p = \int \frac{q}{\sigma^2} e^{(-\frac{q}{2\sigma^2})} dp + A = A - e^{(-\frac{q^2}{2\sigma^2})},$$

$$q = \sqrt{2\sigma^2 |\ln(A - p)|}.$$

E das condições $0 < p < 1$ e $0 < q < \infty$ determinamos a constante de integração A .

$$q = \sqrt{2\sigma^2 |\ln(1 - p)|}. \quad (2.5)$$

De modo que podemos construir o conjunto de vetores de afinidades da forma seguinte:

Geramos um numero aleatório independente p , com distribuição $\rho_2(p)$ uniforme, e de (2.5) obtemos q , uma variável aleatória independente com distribuição dada por (2.4). Finalmente identificamos cada valor q deste modo gerado, com uma componente q_i^μ do vetor \vec{K}^μ ; deste modo: $q_i^\mu \equiv q$.

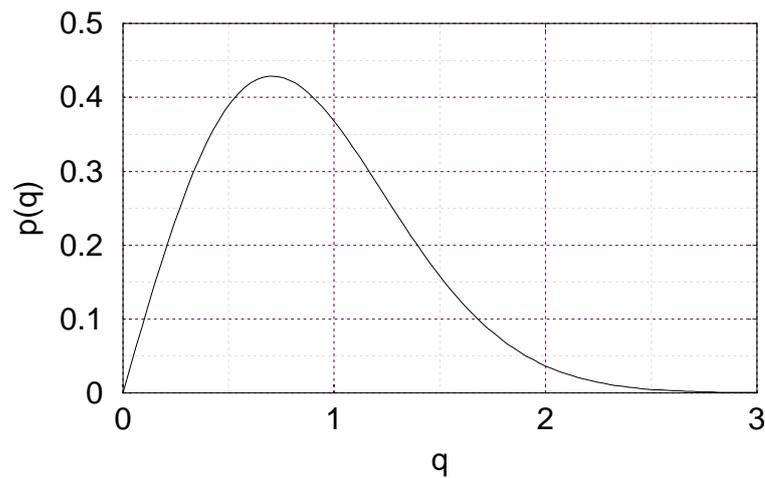


figura 2.3: A função de distribuição 2.4 com $\sigma = 1$ y $\langle q \rangle = \frac{\sqrt{\pi}}{2}$.

2.2.2 Construindo o conjunto de treinamento

Construímos o conjunto de treinamento \mathcal{L} , usando o conjunto de afinidades dado por (2.3) e a equação (2.2).

Para cada odor μ geramos M números aleatório H_ζ^μ independentes e com distribuição dada por $\rho_2(H_\zeta^\mu) = \frac{1}{H_{max}-H_{min}}$ onde: $H_{min} \leq H_\zeta^\mu \leq H_{max}$ e $\zeta = 1.....M$.

Tendo em conta isso, escrevemos a equação (2.2) de forma compacta:

$$S_i^{\mu\zeta} = \frac{q_i^\mu H^{\mu\zeta}}{1 + q_i^\mu H^{\mu\zeta}}, \tag{2.6}$$

vetorialmente fica: $\vec{S}^{\mu\zeta} = \{S_1^{\mu\zeta}.....S_2^{\mu\zeta}\}$, Onde os valores de $O^{\mu\zeta}$ serão:

1 para $\mu = 1$,

-1 para $\mu \neq 1$.

2.2.3 Análise do Conjunto de Treinamento e Mudança da representação

Como q_i^μ e $H^{\mu\zeta}$ são variáveis aleatórias independentes e com distribuição dada por

$$\rho_1(q) = \frac{q}{\sigma^2} e^{-\frac{q^2}{\sigma^2}},$$

e

$$\rho_2(H) = \frac{1}{H_{max} - H_{min}},$$

respectivamente, podemos calcular a distribuição $\rho_4(S_i^{\mu\zeta}) = \rho_4(S)$.

O primeiro é calcular a distribuição $\rho_3(y)$ onde $y = qH$, e donde $\delta(x)$ é a função delta de Dirac temos:

$$\rho_3(y) = \int_0^\infty \int_{H_{min}}^{H_{max}} \delta(y - Hq) \rho_1(q) \rho_2(H) dH dq, \tag{2.7}$$

fazendo $y = Hq$, integrado em y , multiplicando e dividindo por $\sqrt{\pi}$ temos:

$$\rho_3(y) = \frac{\sqrt{\pi}}{\sigma^2(H_{max} - H_{min})} \int_{\frac{y}{H_{max}}}^{\frac{y}{H_{min}}} \frac{2}{\sqrt{\pi}} e^{-\frac{q^2}{\sigma^2}} dq,$$

usando a representação $Erfc(z) = \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$ para a função erro complementar obtemos a relação desejada

$$\rho_3(y) = \frac{\sqrt{\pi}}{\sigma(H_{max} - H_{min})} \left[erfc\left(\frac{y}{H_{max}\sigma}\right) - erfc\left(\frac{y}{H_{min}\sigma}\right) \right]. \quad (2.8)$$

A equação (2.8) é a densidade de probabilidade de y , que tem as propriedades:

$$\int_0^\infty \rho_3(y) dy = 1$$

e

$$\rho_3(y) \geq 0.$$

Como $S(q, H) = S = \frac{y}{1+y}$ podemos agora encontrar a distribuição $\rho(S)$ dada por:

$$\rho(S) = \int_0^\infty \delta\left(S - \frac{y}{1+y}\right) \rho_3(y) dy, \quad (2.9)$$

fazendo a mudança $\frac{y}{1+y} = S$, e aplicando:

$$\int \phi(x) \delta(\psi(x) - x_o) dx = \sum \left. \frac{\phi(x)}{\left| \frac{d\psi(x)}{dx} \right|} \right|_{\psi(x)=x_o},$$

temos:

$$\rho(S) = \frac{\rho_3\left(\frac{S}{1-S}\right)}{(1-S)^2}. \quad (2.10)$$

De (2.8) em (2.10), finalmente obtemos a distribuição desejada:

$$\rho(S) = \frac{\sqrt{\pi}(1-S)^{-2}}{\sigma(H_{max} - H_{min})} \left[Erfc\left(\frac{S}{H_{max}\sigma(1-S)}\right) - Erfc\left(\frac{S}{H_{min}\sigma(1-S)}\right) \right]. \quad (2.11)$$

A equação (2.11) é a densidade de probabilidade de S , que tem as propriedades:

$$0 \leq \rho(S) \leq 1, \quad (2.12)$$

$$\int_0^1 \rho(S) d(S) = 1. \quad (2.13)$$

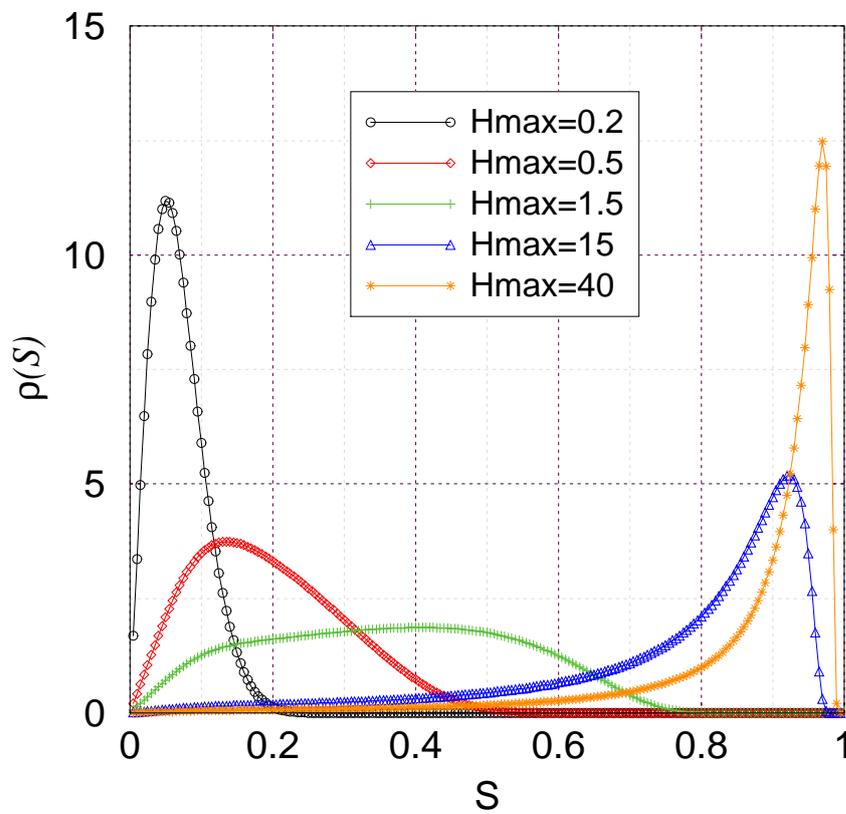


Figura 2.4: A função de distribuição 3.11, com $\sigma = 1$, para $H_{min} = 0.1$ e 5 H_{max} diferentes

Na figura 2.4 e mostrada a função de distribuição $\rho(S)$, vs. a saída dos transdutores, para diferentes valores da concentração máxima H_{max} . Note que o valor esperado $\langle S \rangle$ e a variância de S são função da concentração máxima, no limite de altas

concentrações $S = 1$ com probabilidade 1. Isto sugere fortemente que a capacidade crítica do percéptron (classificador) deve ser uma função da concentração máxima.

Para simplificar o problema de treinamento é conveniente mudar nossa representação do conjunto de treinamento, Isto é: mudaremos a representação do conjunto de treinamento por razões técnicas vinculadas ao algoritmo de aprendizado³. De (2.6) podemos definir nosso conjunto de treinamento como:

$$\mathcal{L} = \{ \vec{S}^{\mu\zeta}, O^{\mu\zeta} \}. \quad (2.14)$$

Onde $O^{\mu\zeta}$ é definido pela natureza do problema que nós estamos considerando. De acordo com isso temos:

$$O^{\mu\zeta} = \begin{cases} 1 & \mu = 1 \\ -1 & \mu \neq 1 \end{cases}, \quad (2.15)$$

$$\mu = 1 \dots P, \quad \zeta = 1 \dots M.$$

E preciso normalizar nossos vetores de entrada⁴.

$$\vec{X}^{\mu\zeta} = \frac{\vec{S}^{\mu\zeta}}{\sqrt{\sum_{i=1}^N (S_i^{\mu\zeta})^2}}, \quad (2.16)$$

e de (2.14) nosso conjunto de treinamento é:

$$\mathcal{L} = \{ \vec{X}^{\mu\zeta}, O^{\mu\zeta} \}, \quad (2.17)$$

$$\vec{X}^{\mu\zeta} = \{ X_1^{\mu\zeta} \dots X_N^{\mu\zeta} \}.$$

Para cada vetor $\vec{X}^{\mu\zeta}$ apresentado ao percéptron, calculamos o campo sináptico $J^{\mu\zeta} = \sum_{i=1}^N X_i^{\mu\zeta} w_i = \vec{X}^{\mu\zeta} \cdot \vec{w}$, e a saída do percéptron será:

$$\sigma^{\mu\zeta} = \text{sgn}(J^{\mu\zeta}), \quad (2.18)$$

onde $\text{sgn}(\theta)$ é:

³A necessidade dessa mudança na representação ficará clara no capítulo 4

⁴Esta normalização é necessária, dada a natureza do algoritmo de treinamento (capítulo 4)

$$\text{sgn}(\theta) = \begin{cases} 1 & \theta \geq 0 \\ -1 & \theta < 0 \end{cases} .$$

É conveniente trocar esta representação por: $[\vec{X}^{\mu\zeta} O^{\mu\zeta}] \cdot \vec{w} > 0$, e por simplicidade fazemos $[\vec{X}^{\mu\zeta} O^{\mu\zeta}] = \vec{\xi}^{\mu\zeta}$ de modo que (2.14) fica finalmente como:

$$\vec{w} \cdot \vec{\xi}^{\mu\zeta} > 0. \quad (2.19)$$

Em adiante chamaremos de $h^{\mu\zeta} = J^{\mu\zeta} O^{\mu\zeta}$ ao campo, definido como $h^{\mu\zeta} = \vec{w} \cdot \vec{\xi}^{\mu\zeta}$.

Note que se para todo μ, ζ o campo $h > 0$, então \vec{w} é o vetor de pesos que resolve o problema, Isto é, se encontro um vetor \vec{w} que satisfaz a condição (2.19) para todo μ, ζ , a este vetor chamaremos de \vec{w}^* .

Ao percéptron serão apresentados os vetores $\vec{\xi}^{\mu\zeta}$, todos eles membros de nosso conjunto de treinamento. O problema de aprendizado consiste em encontrar um vetor de pesos \vec{w}^* que satisfaça a desigualdade (2.19), ou estabelecer que o dito vetor não existe para este específico conjunto de treinamento.

Capítulo 3

O algoritmo de treinamento

3.1 Introdução

No percéptron de Rosenblatt[3] os vetor de pesos \vec{w} são modificados de acordo com a regra:

$$\vec{w}_{new} = \vec{w} + \theta(-h_u)\xi_\mu, \quad (3.1)$$

$$\text{onde } h_\mu = \vec{w} \cdot \vec{\xi} \quad \text{e} \quad \theta(x) = \begin{cases} 1, & x > 0 \\ 0, & x \leq 0 \end{cases}.$$

Um *teorema de convergência* pode ser provado para esta regra de treinamento. Se existe uma solução, o percéptron encontrará o correto vetor de pesos \vec{w} em um finito número de passos. Existem muitas versões do algoritmo de Rosenblatt (Abbott e Kepler 1989; Kruath e Mezard 1987; Anlauf e Biehl 1989; Kinzel and Rujan 1990). Todos esses métodos convergem se uma solução existe, mas não detectam a ausência de solução.

Existem outros tipos de algoritmos que detectam a ausência de uma solução (Ho e Kashyap 1965), mas não são locais e não podem ser realizados por neurônios.

Nabutoovsky e Domany[11] encontraram um algoritmo local que indica se uma solução não existe, em um processo normal de treinamento do percéptron. O algoritmo é inicializado em: $\vec{w} = \frac{\xi_1}{\sqrt{N}}$, $\vec{\xi}_\mu \in B^N$, $\mu = 1 \dots p$, $B = \{-1, 1\}$ $d = \frac{1}{\sqrt{N}}$. Este parâmetro novo, d , será conservado e modificado no transcurso do processo de treinamento. Os exemplos são apresentados ciclicamente, e um erro é encontrado se $J_\mu = \vec{w} \cdot \vec{\xi}_\mu \leq 0$ para algum parâmetro μ . O procedimento de aprendizagem consiste

em modificar os pesos \vec{w} e o parâmetro d de acordo com:

$$\vec{w}_{new} = \frac{\vec{w} + \eta \vec{\xi}_\mu}{|\vec{w} + \eta \vec{\xi}_\mu|}, \quad (3.2)$$

$$d_{new} = \frac{d + \eta}{\sqrt{1 + 2\eta h_\mu + \eta^2 N}}, \quad (3.3)$$

$$\eta = \frac{-h_\mu + 1/d}{N - h_\mu/d}. \quad (3.4)$$

O processo de treinamento termina quando se haja aprendido todos os exemplos, ou quando o parâmetro d alcance seu valor crítico:

$$dc = \frac{N^{(\frac{N+1}{2})}}{2^{N-1}}. \quad (3.5)$$

3.2 O Algoritmo

O algoritmo apresentado neste trabalho é uma extensão do algoritmo de Nabutovsky e Domany[11] para: $\vec{\xi}^{\mu\zeta} \in \{+\Re\}^N$.

O parâmetro d e o vetor de pesos \vec{w} , são primeiramente inicializados a:

$$d = 1, \quad \vec{w} = \vec{\xi}^{11}. \quad (3.6)$$

Apresentamos os exemplos cíclica e repetidamente, um erro é encontrado se:

$$\vec{w} \cdot \vec{\xi}^{\mu\zeta} \leq 0. \quad (3.7)$$

Neste caso modificamos os pesos \vec{w} e o parâmetro d de acordo com:

$$\vec{w} = \frac{\vec{w} + \eta \vec{\xi}^{\mu\zeta}}{|\vec{w} + \eta \vec{\xi}^{\mu\zeta}|}, \quad (3.8)$$

$$d_{new} = \frac{d + \eta}{\sqrt{1 + 2\eta h^{\mu\zeta} + \eta^2}}, \quad (3.9)$$

onde

$$\eta = \frac{1 - dh^{\mu\zeta}}{(d - h^{\mu\zeta})}. \quad (3.10)$$

E processo de aprendizado é terminado quando a condição (2.19) é satisfeita, o que significa que uma solução foi encontrada. Alternativamente o processo é abortado se o parâmetro d excede o valor crítico dado por:

$$d > d_c = \sqrt{N} [N G_{max}]^{\frac{N}{2}} |\det B|, \quad (3.11)$$

Onde

$$G_{max} = \max \left[\sum_i (S_i^{\mu\zeta})^2 \right], \quad (3.12)$$

e $B = [b_{ij}]_{i,j=1}^N$ é uma matriz $N \times N$, construída da escolha de N linhas da matriz $\vec{X}^{\mu\zeta}$. Neste caso o problema não tem solução.¹

3.3 Análises do algoritmo

Desejamos Provar que se não existe uma solução, a condição (3.11) é satisfeita. Para ver isto, provaremos que:

$$d \leq \frac{\vec{w} \cdot \vec{w}^*}{h_{min}^*} \leq \frac{1}{h_{min}^*} \leq d_c. \quad (3.13)$$

Primeiro assumimos que uma solução existe, e de (2.19) devemos ter:

$$\vec{w}^* \cdot \vec{\xi}^{\mu\zeta} > 0, \quad (3.14)$$

onde \vec{w}^* é uma solução de (2.19)

De (2.16) os vetores de entrada do percéptron $|\vec{\xi}^{\mu\zeta}| = 1$ são normalizados, e da condição (3.6) o vetor de pesos será inicialmente $\vec{w} = \vec{\xi}^{\vec{11}}$, de forma que $|\vec{w}| = \sqrt{\sum_{i=1}^N (w_i)^2} = 1$. Esta normalização será preservada no processo de aprendizagem equação (3.8).

O primeiro passo é notar que se uma solução existe devemos ter que:

¹O conjunto de treinamento não é linearmente separável.

$$\sum_{i=1}^N (w_i w_i^*) = \cos(\psi) \leq 1. \quad (3.15)$$

O segundo passo é provar que a condição $\frac{\vec{w} \cdot \vec{w}^*}{h_{min}^*} \geq d$ é satisfeita, onde $h_{min}^* = \min\{h_{\mu\zeta}^* = \vec{w}^* \cdot \vec{\xi}^{\mu\zeta}\}$. E o terceiro passo é provar que se (2.19) tem solução, então $h_{min} \geq \frac{1}{d_c}$.

3.3.1 Prova de $\frac{\vec{w} \cdot \vec{w}^*}{h_{min}^*} \geq d$

De (3.6), começamos o processo de aprendizagem com $d = 1$, e segue que:

$$\sum_{i=1}^N w_i w_i^* = \vec{w}^* \cdot \vec{\xi}^{11} = h_{11}^* \geq h_{min}^* = h_{min}^* d, \quad (3.16)$$

$$\frac{\vec{w} \cdot \vec{w}^*}{h_{min}^*} \geq d. \quad (3.17)$$

Para algum estado intermédio, e depois de um passo de treinamento, temos:

$$\vec{w}_{new} \cdot \vec{w}^* = \frac{(\vec{w} + \eta \vec{\xi}^{\mu\zeta}) \cdot \vec{w}^*}{\sqrt{(\vec{w} + \eta \vec{\xi}^{\mu\zeta})^2}} = \frac{\vec{w} \cdot \vec{w}^* + \eta h_{\mu\zeta}^*}{\sqrt{1 + 2h_{\mu\zeta} \eta + \eta^2}},$$

e da equação (3.17)

$$\vec{w}_{new} \cdot \vec{w} \geq \frac{h_{min}^* (d + \eta)}{\sqrt{1 + 2\eta h_{\mu\zeta} + \eta^2}},$$

onde fazemos

$$\frac{d + \eta}{\sqrt{1 + 2\eta h_{\mu\zeta} + \eta^2}} = d_{new},$$

Finalmente nós concluímos que:

$$\vec{w}_{new} \cdot \vec{w} \geq h_{min}^* d_{new}. \quad (3.18)$$

Isto prova $\frac{\vec{w} \cdot \vec{w}^*}{h_{min}^*} \geq d$ para todo \vec{w} e d , no transcurso do processo de treinamento.

3.3.2 Prova de $h_{min}^* \geq \frac{1}{d_c}$

Se (2.19) tem uma solução, o sistema:

$$\vec{\xi}^{\mu\zeta} \cdot \vec{U} \geq 1, \quad (3.19)$$

tem também uma solução, onde \vec{U} não está normalizado. Por definição, ao menos N desigualdades linearmente independentes de (3.19) serão satisfeitas como igualdades:

$$\xi_i^{\mu\zeta} \vec{U}^* = 1 \quad i = 1, \dots, N. \quad (3.20)$$

Da regra de Cramer's a solução de (3.20) está dada por:

$$U_i^* = \frac{\det A^{(i)}}{\det A} \quad i = 1, \dots, N, \quad (3.21)$$

onde $A = \vec{\xi}_i^{\mu\zeta}$ é obviamente uma matriz $N \times N$, e $A^{(i)}$ é a matriz obtida do processo de substituição de todos os elementos da coluna i por 1.

Para qualquer matriz quadrada $A = [a_{ij}]_{i,j=1}^N$, pode-se mostrar a partir da desigualdade de Cauchy-Schwartz que:

$$|\det A|^2 \leq \prod_{i=1}^N \sum_{j=1}^N [|a_{ij}|^2], \quad (3.22)$$

conhecida como a desigualdade de Hadamard's [12]

Aplicando (37) a (37)

$$|\det A^i| \leq \left[\prod_{i=1}^N \sum_{j=1}^N |a_{ij}|^2 \right]^{\frac{1}{2}} \leq N^{\frac{N}{2}}. \quad (3.23)$$

Notemos que a matriz A em (3.21) tem a forma:

$$A = \begin{vmatrix} \frac{b_{11}}{[G_1]^{1/2}} & \cdot & \cdot & \frac{b_{1N}}{[G_1]^{1/2}} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ \frac{b_{N1}}{[G_N]^{1/2}} & \cdot & \cdot & \frac{b_{NN}}{[G_N]^{1/2}} \end{vmatrix},$$

$$G_i = \sum_{j=1}^N [b_{ij}]^2 \quad i = 1 \dots N, \quad (3.24)$$

de modo que podemos escrever o determinante de A como:

$$|\det A| = \frac{|\det B|}{\prod_{i=1}^N [G_i]^{1/2}}, \quad (3.25)$$

onde $B = [b_{ij}]_{i,j=1}^N$. De (3.22) escreve-se (3.25) como:

$$|\det A| \geq \frac{|\det B|}{[\max G_i]^{N/2}}. \quad (3.26)$$

Da equação (3.21), (3.23) e (3.26) escrevemos:

$$|U_i^*| \leq \frac{[N \max G_i]^{N/2}}{|\det B|},$$

e então:

$$|\vec{U}^*| \leq \frac{\sqrt{N} [N \max G_i]^{N/2}}{|\det B|}, \quad (3.27)$$

onde o $|\det B|$ pode ser obtido computacionalmente. A figura 3.1 mostra o comportamento típico do modulo do determinante de B para o tipo de dados de nosso conjunto de treinamento.

$$|\det B| \approx \exp\left[-\frac{N}{2}\right].$$

Finalmente fazendo $\vec{w}^* = \frac{\vec{U}^*}{|\vec{U}^*|}$, e usando (3.19) com (3.27), e da definição (3.11) podemos ver que:

$$\vec{w}^* \cdot \vec{\xi}^{\mu h} \geq \frac{1}{\sqrt{N} [N \max G_i]^{N/2} |\det B|} = \frac{1}{d_c}. \quad (3.28)$$

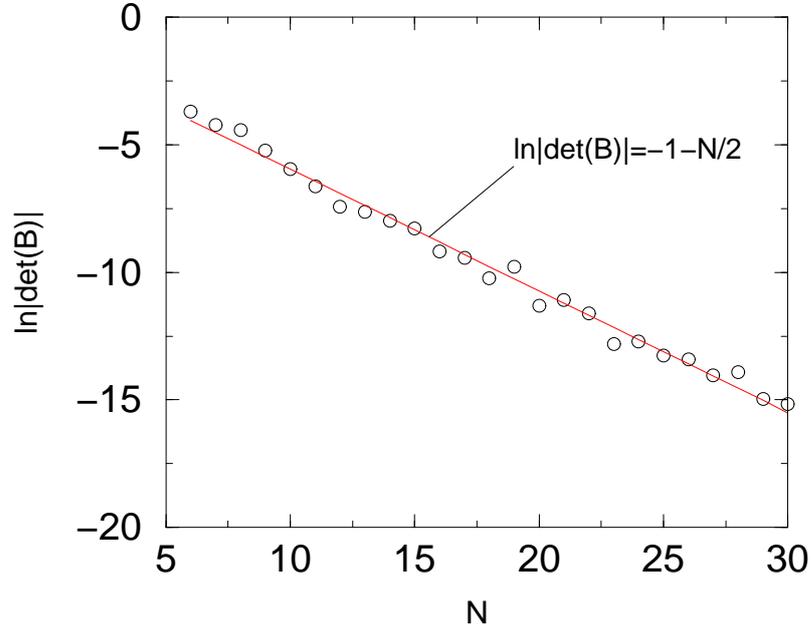


Figura 3.1: Gráfico do $\ln |\det B|$, como função do número de linhas ou colunas. Os elementos de B são gerados aleatoriamente e com distribuição dada por (2.11). O gráfico mostra a média do $\ln |\det B|$ de 100 matrizes B .

3.4 Teorema de Convergência

Desejamos mostrar que o algoritmo termina o processo de treinamento depois de um finito número de passos.

Escrevemos $d_{new} = d_{new}[\eta_{opt}(h_{\mu\zeta}), h_{\mu\zeta}]$ onde η_{opt} é o valor ótimo de η que faz com que d_{new} seja máximo, o que fazemos tomando $\left(\frac{\partial d_{new}}{\partial \eta}\right)\Big|_{\eta_{opt}} = 0$. Disso obtemos

$$\eta_{opt}(h_{\mu\zeta}) = \frac{1 - dh_{\mu\zeta}}{(d - h_{\mu\zeta})},$$

para $h_{\mu\zeta} < 0$ (ver figura 3.2), teremos:

$$d_{new} = d_{new}[\eta_{opt}(h_{\mu\zeta}), h_{\mu\zeta}] \geq d_{new}\left(\frac{1}{d}, h_{\mu\zeta}\right),$$

$$\geq d_{new}\left(\frac{1}{d}, 0\right) = \frac{d + \frac{1}{d}}{\sqrt{1 + \frac{1}{d^2}}} = \sqrt{d^2 + 1},$$

$$d_{new}^2 \geq d^2 + 1. \quad (3.29)$$

Se iteramos a equação (3.29) podemos ver que depois de d_c^2 passos, d poderia exceder a d_c .

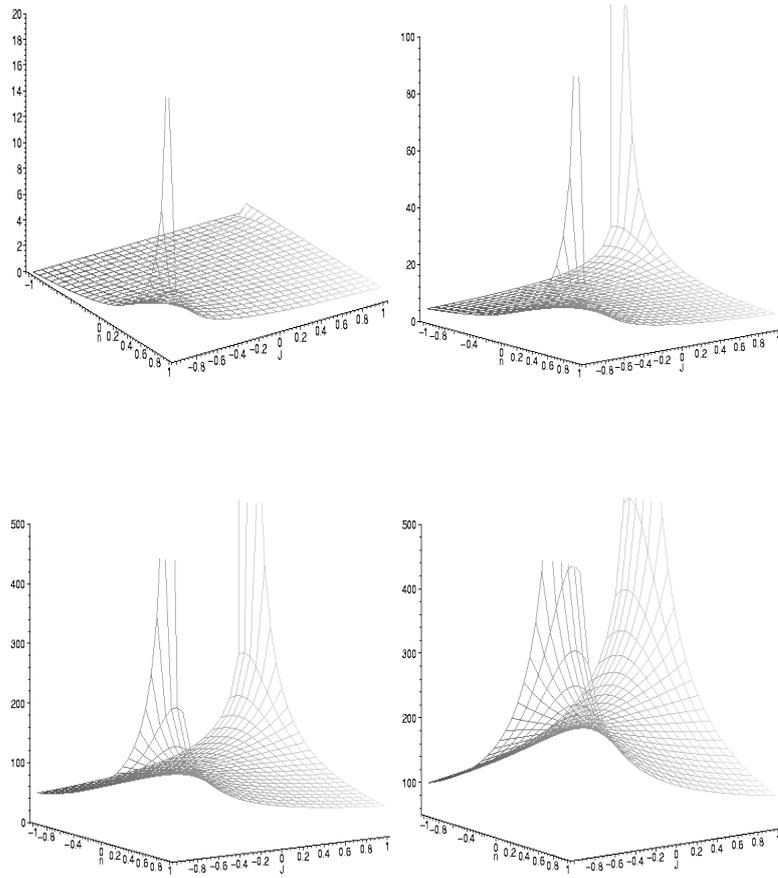


figura 3.2: O parâmetro d_{new} em função de η e o campo $h_{\mu\zeta}$, para 4 diferentes d

A figura 3.2 mostra como, se existe uma solução, o campo é confinado a um intervalo cada vez mais estreito em torno de $h = 0$, enquanto que o parâmetro d fica crescendo linearmente. Se não existe solução, o campo não poderá ser confinado, e como se computa η de modo que se maximize o parâmetro d , este ficará crescendo exponencialmente.

3.5 O crescimento exponencial de d

Estudemos agora o caso mais significativo, em que a equação (2.19) não tem solução. Neste caso, o parâmetro d crescerá exponencialmente. A prova de elo será mostrada nesta seção.

Definição:

$$h_o = - \max_{|U|=1} \min_{\mu} U \cdot \xi^{\mu\zeta}. \quad (3.30)$$

Se o sistema $W \cdot \xi^{\mu\zeta} \geq 0$ não tem solução, $h_o > 0$.

Da definição de h_o , $\min_{\mu} W \cdot \xi^{\mu\zeta} \leq -h_o$. Deste modo, iniciando o processo de aprendizado existe um parâmetro ν para o qual $h_{\nu} \leq -h_o$.

Os passos de treinamento dados por (3.8) são tais que depois de processar o parâmetro ν temos $h_{\nu} \geq 0$. Denotemos por $h_i = h_{\nu}$ quando $h_{\nu} \leq h_o$ e por $h_f = h_{\nu}$ quando $h_{\nu} \geq h_o$. Agora estabelecemos um limite no incremento de d durante os processos de aprendizado entre os instantes i e f . Para isto notemos primeiro que depois de qualquer aplicação de 3.7,

$$d_{new} = d_{new}[\eta_{opt}(h_{\mu\zeta}), h_{\mu\zeta}] \geq d_{new}(-h_{\mu\zeta}, h_{\mu\zeta}), \quad (3.31)$$

onde a função d_{new} e η_{opt} são definidos por (3.9) e (3.10). Substituindo-se (3.9) em (3.31) e usando $h_{\mu\zeta} < 0$, obteremos $d_{new} \geq \frac{d}{\sqrt{1-h_{\mu\zeta}^2}}$. Esta desigualdade se pode escrever como:

$$\ln d_{new} \geq \ln d + \frac{h_{\mu\zeta}^2}{2}, \quad (3.32)$$

somando essa desigualdade sobre todos os passos de treinamento, desde i até f , obtemos

$$\ln d_f - \ln d_i \geq \sum_{\mu, \zeta} \frac{h_{\mu\zeta}^2}{2}, \quad (3.33)$$

essa soma corre sobre todos os μ e ζ nos instantes i e f , nos quais os passo de treinamento não são triviais. Concentremos agora no valor de h_ν durante esses passos de treinamento. De (3.8) e (3.9):

$$\begin{aligned} \left(\frac{h_{\mu\zeta}}{d}\right)_{new} &\leq \frac{h_\nu + \eta}{d + \eta}, \\ \left(\frac{h_\nu}{d}\right)_{new} - \left(\frac{h_\nu}{d}\right) &\leq \left(\frac{1}{d + \eta} - \frac{1}{d}\right) + \frac{\eta}{d + \eta} \leq \frac{-h_\nu \eta}{d^2} + \frac{\eta}{d}. \end{aligned} \quad (3.34)$$

Notemos que entre i e f teremos $h_i \leq h_\nu \leq 0$, e que $d \geq d_i$. Usando isto em (3.34) temos:

$$\left(\frac{h_\nu}{d}\right)_f - \left(\frac{h_\nu}{d}\right)_i \leq \frac{-h_i \eta}{d_i^2} + \frac{\eta}{d_i}, \quad (3.35)$$

por definição de f e i ,

$$\left(\frac{h_\nu}{d}\right)_f - \left(\frac{h_\nu}{d}\right)_i \geq \frac{-h_i}{d_i}, \quad (3.36)$$

somamos agora todos os passos não triviais entre i e f , e usando (3.36), conseguimos:

$$\sum_{\mu\zeta} \eta_{\mu\zeta} \left(\frac{-h_i}{d_i^2} + \frac{1}{d_i}\right) \geq \frac{-h_i}{d_i}.$$

Usando $d_i \geq 1$ e $0 \leq -h_i \leq 1$, conseguimos:

$$\sum_{\mu\zeta} \eta_{\mu\zeta} \geq \frac{-h_i}{2}, \quad (3.37)$$

pela definição (3.10), a soma está dada por

$$\sum_{\mu\zeta} \eta_{\mu\zeta} = \sum_{\mu\zeta} \frac{-h_{\mu\zeta} + 1/d_{\mu\zeta}}{1 - h_{\mu\zeta}/d_{\mu\zeta}} \leq \sum_{\mu\zeta} \left[-h_{\mu\zeta} + \frac{1}{d_i}\right], \quad (3.38)$$

combinando (3.37) e (3.38) temos

$$-\sum_{\mu\zeta} h_{\mu\zeta} \geq \frac{-h_{\mu\zeta}}{2} - \frac{P_e}{d_i} \geq \frac{h_o}{2} - \frac{P_e}{d_i}, \quad (3.39)$$

onde $P_e \leq P$ é o número de termos na soma. Consideremos agora 2 casos.

Primeiro quando $d_i \geq \frac{4P}{h_o}$, usamos (3.39) para estabelecer um limite inferior em $-\sum_{\mu\zeta} h_{\mu\zeta} \geq \frac{h_o}{4}$. Isto gera um limite inferior em

$$\sum_{\mu\zeta} h_{\mu\zeta} \geq P_e (h_o/4P_e)^2 \geq \frac{h_o^2}{16P},$$

usando (3.33)

$$\ln d_f - \ln d_i \geq \frac{h_o^2}{32P}. \quad (3.40)$$

Portanto, em qualquer passo, $\ln d$ se incrementa num fator de $\frac{h_o^2}{32P}$, e da condição $d \geq 1$ concluímos que depois de

$$N_s = \frac{32P}{h_o^2} \ln(d_c). \quad (3.41)$$

passos, a condição (3.11) será alcançada.

No segundo caso, $1 \leq d_i \leq 4P/h_o$, é fácil ver, da condição (3.29), que depois de $[\frac{4P}{h_o}]^2$ o parâmetro d se incrementa até alcançar a condição em que o caso prévio é aplicável. Conseqüentemente, este número de passos tem que ser somado ao número acima calculado.

3.6 Simulação

Testamos o algoritmo com o conjunto do treinamento dado por (2.19), com $N = 10$, $H_{max} = 50$ e $P = 25$. Este valor $\alpha = p/N$ está na região considerada como “difícil” de aprender². Os resultados obtidos são apresentados na figura 3.3. O caso em que uma solução é encontrada é mostrado em la figura 3.3-A, e o casos em que não existe solução é mostrado na figura 3.3-B.

²Isto não é exatamente certo, posto que não conhecemos a capacidade crítica do percéptron neste problema. No entanto é razoável pois a capacidade crítica do percéptron simples é 2.

Quando a solução existe d^2 incrementa-se linearmente de acordo com (3.29). Cada linha corresponde a um conjunto de treinamento particular. O processo termina quando uma solução é encontrada.

A figura 3.3b mostra o parâmetro d . Quando o problema não é linearmente separável, o parâmetro d se incrementa exponencialmente de acordo com (3.41). Cada linha corresponde a um conjunto de treinamento particular. O processo termina quando d alcança d_c .

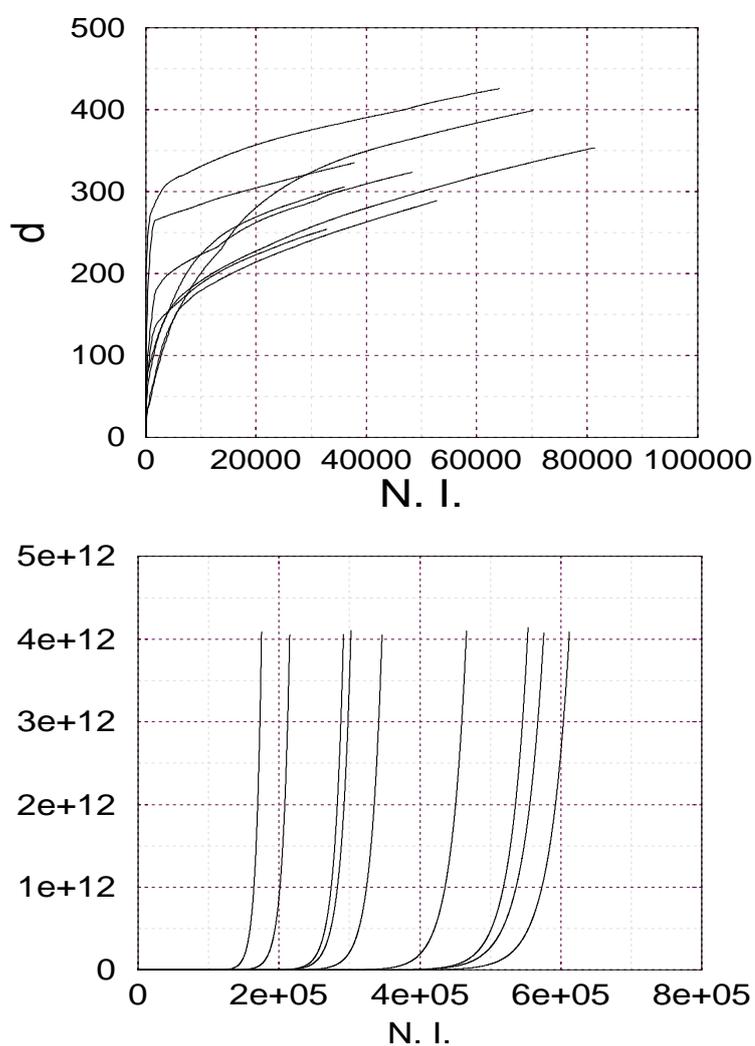


Figura 3.3: Parâmetro b vs. o número de passos de treinamento onde $N=10$ e $P=25$, (a) Caso linearmente separável (b) caso linearmente não

separável. Cada linha corresponde a um experimento.

Se o conjunto de treinamento não é linearmente separável e $\alpha = \frac{p}{N} \gg \alpha_c$, o processo do treinamento não é demorado, contrariamente se o conjunto de treinamento é linearmente separável e $\alpha = \frac{p}{N} \ll \alpha_c$, então, o algoritmo encontra rapidamente uma solução. Na região em que $\alpha \approx \alpha_c$, o processo de treinamento é muito mais demorado. (O número de passos de treinamento para alcançar quaisquer das duas condições depende do quão perto do α_c estejamos.)

É interessante comparar o valor do parâmetro d quando o perceptron aprende e quando não aprende. A diferença entre $d_{aprende}$ e $d_{n.aprende}$ é extremamente alta. $d_{aprende}$ em todos os casos testados é sempre da ordem de: $d_{aprende} < 1000$ para $N \leq 10$. No entanto, $d_{n.aprende}$ é de ordem de: $d_{n.aprende} \approx 1 \times 10^{12}$. Isso sugere claramente que não é preciso chegar tão longe do valor de d para deter o processo, quando o problema é não linearmente separável.

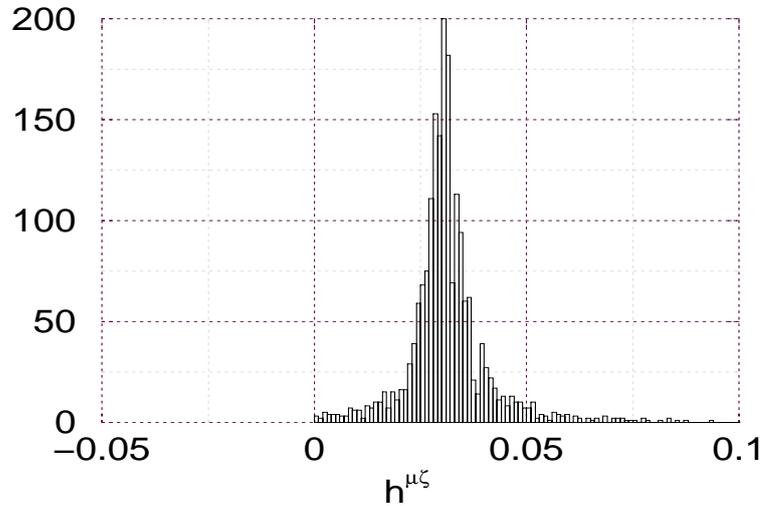


Figura 3.4-A: Histograma de $\vec{w} \cdot \vec{\xi}^{\mu\zeta} = h^{\mu\zeta}$, todos os vetores foram corretamente classificados, o conjunto de treinamento é linearmente separável.

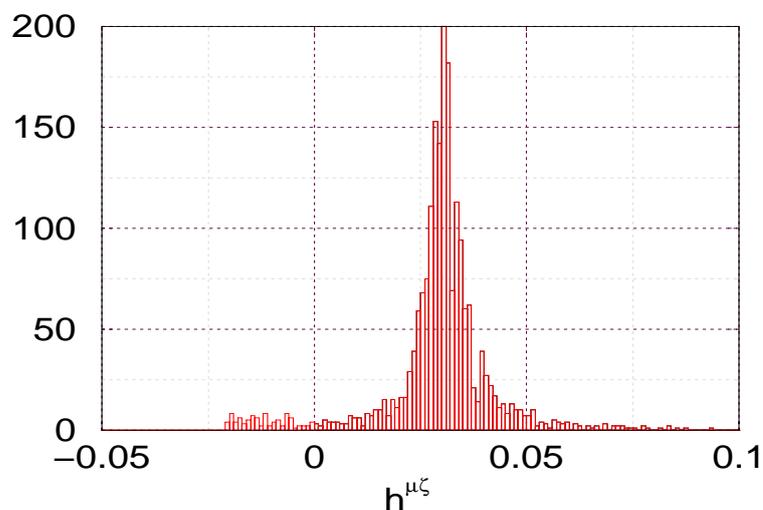


Figura 3.4-B: Histograma de $\vec{w} \cdot \vec{\xi}^{\mu\zeta} = h^{\mu\zeta}$. A maioria dos vetores foram corretamente classificados, no entanto o problema não tem solução. O conjunto de treinamento é linearmente não-separável

A figura 3.4 mostra o histograma do $\vec{w} \cdot \vec{\xi} = h^{\mu\zeta}$, depois de terminado o processo de aprendizado (o processo de treinamento é terminado quando $d \geq d_c$ ou a condição (3.7) é satisfeita) onde o conjunto de treinamento é construído com $M_1 = 100$, $M_{\zeta \neq 1} = 100$, $N = 10$, $p = 30$, daí a conclusão de que o conjunto de treinamento tem 3000 vetores. A figura 3.4-A mostra o problema linearmente separável. Todos os vetores foram classificados corretamente. A figura 3.4-B mostra o problema linearmente não separável. Note que uma fração dos vetores não foram classificados corretamente. O problema não tem solução: Não importa o número de passos de treinamento, o número de vetores mal classificados não poderá ser muito menor.

3.6.1 Caso Linearmente Separável

Chamemos de X_b aqueles vetores que não foram classificados corretamente em algum passo no processo de treinamento. Então $h_b = W \cdot X_b$ é o campo devido àqueles

vetores que ainda não foram corretamente classificados.

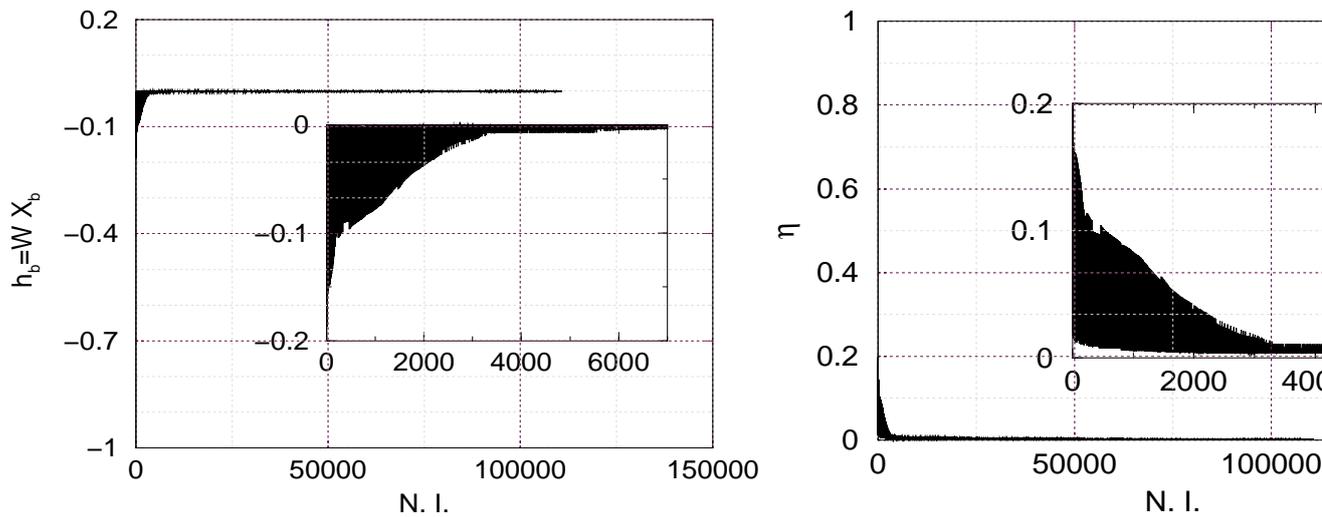


Figura 3.5: O campo h_c e o parâmetro η quando o conjunto de treinamento é linearmente separável. (a) O campo devido aos exemplos que ainda não foram corretamente classificados vs. o número de passos. (b) O parâmetro η vs. o numero de passos.

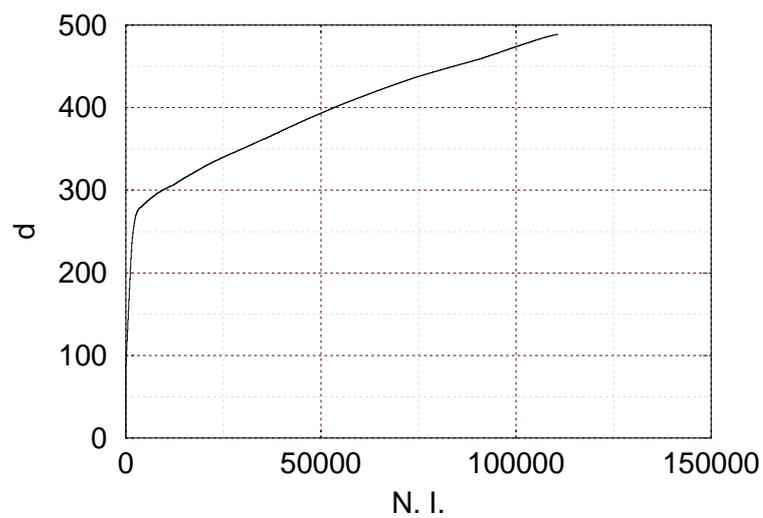


Figura 3.6: O parâmetro d , vs. o número de passos de treinamento quando o conjunto é linearmente separável. Note que d se incrementa linearmente.

Quando o algoritmo é capaz de “aprender” o conjunto de treinamento, é preciso um número exponencialmente grande de passos para atingir d_c . A razão disto é que o campo fica decrescendo, enquanto que o parâmetro η decresce também figura³ (3.5-A 3.5-B). É interessante notar que o parâmetro d traça uma curva que sempre é convexa quando o algoritmo é capaz de encontrar uma solução. Este importante fato pode ser usado na prática como critério para parar o processo de treinamento antes de atingir o valor de d_c .

3.6.2 Caso Linearmente Não separável

Quando o algoritmo não é capaz de encontrar uma solução, o campo h_b , depois de um número de passos de treinamento, não pode ser mais modificado (figura 3.7-A).

³Notemos que o parâmetro η é um reflexo do campo h

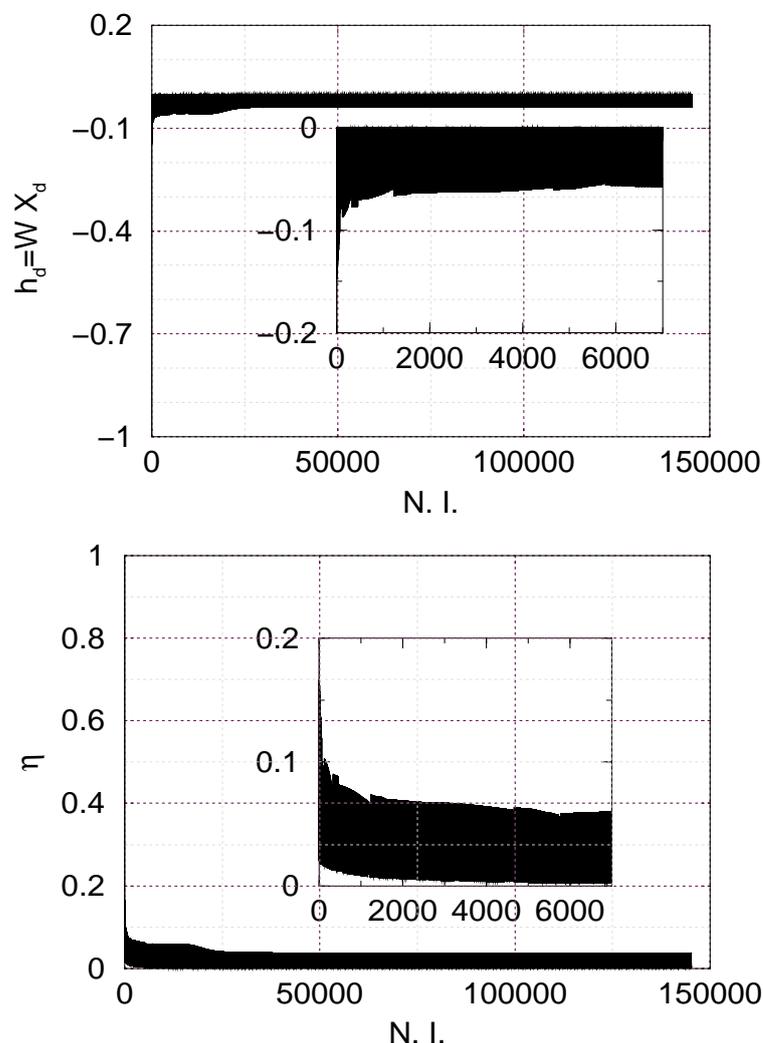


Figura 3.7: O campo e o parâmetro η quando o conjunto de treinamento é linearmente não separável. (a) O campo devido aos exemplos que ainda não foram corretamente classificados vs. o número de passos. (b) O parâmetro η vs. o número de passos.

Enquanto em cada passo de treinamento o parâmetro d fica aumentando, e aquela região no plano $\eta - h$ em que o parâmetro d cresce linearmente fica cada vez mais estreita, eventualmente o parâmetro d crescerá exponencialmente (figura 3.8), devido a que computamos o valor de η de modo que maximize o parâmetro d .

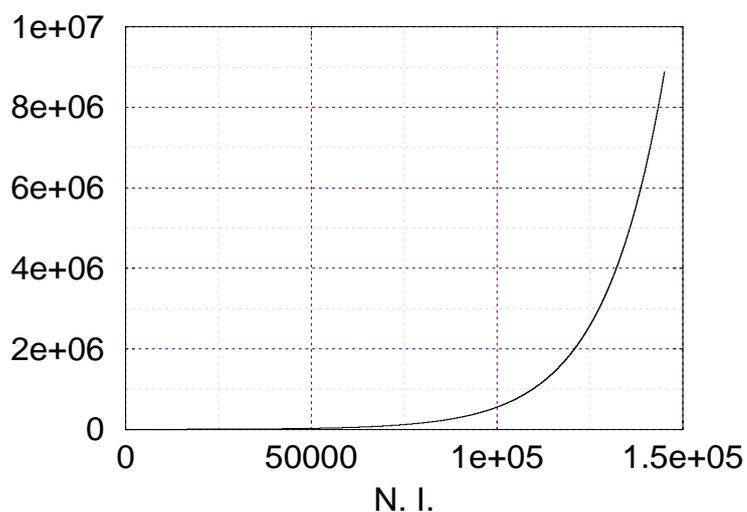


Figura 3.8: O parâmetro d , vs. o número de passos de treinamento quando o conjunto de treinamento é linearmente separável. Note que d cresce exponencialmente

Note o comportamento do campo h_b em ambos casos. A diferença é óbvia: quando o problema tem solução o campo h_b fica decrescendo em cada iteração relevante do algoritmo; quando o problema não tem solução, h_c decresce também, mas só até um ponto bem definido, a partir do qual não decresce mais. Neste ponto, (ver figura 3.7) o algoritmo “percebe” que o campo não cresce mais e o parâmetro d começa a crescer exponencialmente. Isso acontece na figura 3.8 ao redor da iteração relevante igual a 50000. Note na figura 3.7 que o campo pára de decrescer quando se atinge mais ou menos 50000 iterações relevantes.

3.7 Conclusão

Em todas as simulações feitas, observamos que quando o problema é linearmente separável, d_c nunca excede a 1000 ($d_c < 1000$, para $N \leq 10$), enquanto que, quando o problema é não separável, estamos considerando um d_c extremamente grande. Isto nos permite garantir que os resultados estão certos. Ouseja: Isto significa que

um conjunto de treinamento linearmente separável será classificado como aprendível praticamente em todos os casos, e do mesmo modo, um conjunto de treinamento não linearmente separável será classificado como não aprendível praticamente em todos os casos.

Na prática, o valor de d_c dado pela equação (3.11) não pode ser usado como critério para parar o processo de treinamento quando o valor de N é grande⁴. A razão disto é que para valores grandes de N , d_c é exponencialmente grande, e o número de passos de treinamento para atingir este valor é computacionalmente irrealizável, pelo tempo de processamento envolvido.

É preciso considerar alternativas para deter o processo de aprendizado quando o problema não tem solução. Uma possibilidade é estudar a curvatura das curvas traçadas pelo parâmetro d .

Note que quando o problema não é linearmente separável, o parâmetro d desenha uma linha que é sempre côncava. Por outro lado (figura 3.8), depois das primeiras 25000 iterações, o campo devido aos exemplos ainda não classificados corretamente não sofre mudança. Depois de observar o campo nas próximas 25000 iterações, concluímos que o campo não poderá ser mais ajustado de modo que o número de erros decresça, portanto poderíamos terminar aqui o processo. Como vemos, não é preciso fazer mais de 50 mil iterações; no entanto, no gráfico são mostradas 150 mil iterações, sem ainda d alcançar d_c .

Neste trabalho, estudamos perceptron de dimensão restrita a $N \leq 30$, de modo que podemos ainda usar o d_c estabelecido no capítulo 3.

⁴Chamamos $N > 30$ de grade neste trabalho

Capítulo 4

Simulação

Trataremos, neste capítulo, do problema da capacidade de um percéptron para separar odores.

O problema da capacidade no percéptron para valores de entrada aleatórios e contínuos está resumido na pergunta:

Quantos pares de entrada-saída podem confiavelmente ser armazenados num percéptron de dimensões dadas?

A resposta a esta pergunta para entradas aleatórias contínuas que foi derivada por Cover[13], é simples. $p_{max} = 2N$, onde N é o número de entradas do percéptron e é supostamente grande; p_{max} é o número de vetores que o percéptron pode armazenar.

O cálculo de Cover da máxima capacidade de armazenamento de um percéptron é interessante, mas é de difícil generalização. E. Gardner[14] estabeleceu uma nova abordagem para o estudo de redes neurais, que diz respeito à dinâmica das conexões sinápticas em detrimento do tradicional enfoque sobre a dinâmica dos neurônios. Deste ponto de vista, busca-se calcular o volume do conjunto de conexões sinápticas, no espaço de conexões, que classificaria corretamente um certo conjunto de padrões utilizados no treinamento da rede. Essa medida é definida como a capacidade de armazenamento ou capacidade de memorização de um percéptron. Gardner introduziu uma energia da forma:

$$H[w] = \sum_{i=1}^N \sum_{\mu=1}^p \Theta(\kappa - \gamma_i^\mu[w]), \quad (4.1)$$

associada ao número de erros cometidos pela rede na classificação, onde $\gamma_i^\mu[w] = S_i^\mu \sum_{j=1}^N w_{ij} S_j^\mu / \|w_i\| > \kappa$ $\|w_i\| = \sqrt{\sum_j w_{ij}^2}$. A função de partição no ensemble canônico se pode escrever como:

$$Z = \int (\prod_{ij} dw_{ij}) \rho(w) e^{-\beta H[w]}, \quad (4.2)$$

onde $\beta = \frac{1}{T}$ e $\rho(w)$ é a densidade de estados no espaço de conexões sinápticas. Uma grandeza interessante é a energia livre $F = -\beta^{-1} \ln Z$. A energia média pode ser calculada de $E = -\frac{\partial \ln Z}{\partial \beta}$. No limite de $T \rightarrow 0$, E é o número mínimo de erros de classificação cometidos por um percéptron. Neste limite, a função de partição fica como:

$$Z = \int (\prod_{ij}^N dw_{ij}) \rho(w) \prod_{i=1}^N \prod_{\mu=1}^P \Theta(\gamma_i^\mu - \kappa). \quad (4.3)$$

A entropia e o número de estados com energia zero, está dado pela formula do ensemble microcanônico

$$s = \sum_{i=1}^N \ln \Omega_i, \quad (4.4)$$

com:

$$\Omega_i = \int (\prod_j^N dw_{ij}) \rho[w] \prod_{\mu=1}^P \Theta(\gamma_i^\mu[w] - \kappa), \quad (4.5)$$

usando réplicas $\langle \langle \ln \Omega_i \rangle \rangle = \lim_{n \rightarrow 0} \frac{\langle \langle (\Omega_i)^n \rangle \rangle - 1}{n}$, especificando a geometria do espaço de interações $\rho[w] = \frac{\delta(\sum_j w_{ij}^2 - N)}{\int (\prod_j dw_{ij}) \delta(\sum_j w_{ij}^2 - N)}$ (modelo esférico), o parâmetro de Edwards-Anderson $q_{ab} = \frac{1}{N} \sum_{j=1}^N w_{ij}^a w_{ij}^b$ e introduzindo as representações para a função de passo e delta de Dirac:

$$\Theta(\gamma_i^\mu[w^a] - \kappa) = \int_{\kappa}^{\infty} d\lambda_\mu^a \delta(\gamma_i^\mu[w^a] - \lambda_\mu^a), \quad (4.6)$$

$$\delta(q_{ab} - \frac{1}{N} \sum w_{ij}^a w_{ij}^b) = \int_{-\infty}^{\infty} \frac{dF_{ab}}{2\pi/N} e^{iN F_{ab} (q_{ab} - \frac{1}{N} \sum_j w_{ij}^a w_{ij}^b)}, \quad (4.7)$$

$$\delta\left(\sum_j (w_{ij}^a)^2 - N\right) = \int_{-\infty}^{\infty} \frac{dE_a}{2\pi} e^{iE_a (\sum_j (w_{ij}^a)^2 - N)}. \quad (4.8)$$

Essas integrais são manejáveis no limite termodinâmico $N \rightarrow \infty$, usando o método do ponto de sela e impondo a condição de réplicas simétricas:

$$q_{ab} = q \quad F_{ab} = F \quad E_a = a,$$

finalmente conseguimos o valor esperado da entropia:

$$\langle\langle s \rangle\rangle = N^2 \left(\alpha \int_{-\infty}^{\infty} Dy \ln H(\tau) + \frac{1}{2} \ln(1-q) + \frac{1}{2} \frac{q}{1-q} \right), \quad (4.9)$$

onde $Dy = \frac{dy}{\sqrt{2\pi}} e^{-y^2/2}$, $\tau = \frac{\sqrt{q}y + \kappa}{\sqrt{1-q}}$, $H(x) = \frac{1}{2} \text{Erfc}\left(\frac{x}{\sqrt{2}}\right)$.

No limite de $q \rightarrow 1$, o resultado para $\alpha_c = \alpha_c(q=1, \kappa)$ é:

$$\alpha_c^{-1} = \int_{-\kappa}^{\infty} Dy (y - \kappa)^2. \quad (4.10)$$

Para parâmetros de estabilidade aceitável pequenos, $\kappa = 0$, a integral é analiticamente solúvel com $\alpha_c = 2$, precisamente o resultado obtido por Cover.

Formulações similares podem ser propostas para o problema de determinar a máxima capacidade de um percéptron em separar odores. A pergunta neste caso seria:

Quantos odores poden ser apresentados a um percéptron de modo que este possa distinguir um odor do resto, sendo que a discriminação deve ocorrer independente da concentração do odorante?

A determinação analítica deste limite deveria ser abordada com as técnicas de espaço de interações desenvolvidas por Gardner. No entanto, o problema da solução analítica se mostra extremamente complexo pelo alto grau de correlação que o conjunto de treinamento exhibe.

Dado este panorama uma abordagem computacional é relevante, não só do ponto de vista biológico como do ponto de vista de engenharia. Acreditamos que possa mostrar a direção de uma possível solução analítica.

4.1 Escolha dos parâmetros de treinamento

Uma vez que o problema de construir o conjunto de treinamento e de estabelecer o algoritmo ficam resolvidos, o próximo passo é estabelecer as condições apropriadas da simulação. Dois parâmetros são especialmente importantes:

1. Escolher uma concentração mínima, H_{min} .
2. Estabelecer o número apropriado de concentrações por odorante, M .

O ponto 2 é especialmente relevante e será estudado mais detalhadamente na seção (3.1.1). No ponto 1, a escolha de uma concentração mínima pode ser feita estudando-se a equação de resposta dos transdutores $S(q, H)$. Para baixas concentrações S , é linear, portanto pode ser feita qualquer escolha de H_{min} contanto que essa seja razoavelmente pequena (esteja na região linear) e seja diferente que zero. Não se pode escolher um $H_{min} = 0$, posto que isso não tem sentido no contexto do problema.¹ Neste trabalho uma escolha de $H_{min} > 0.5$ é considerada razoável.

4.1.1 O número de concentrações por odorante

É preciso estabelecer um critério para determinar o número apropriado de concentrações M por odorante μ que serão usadas na construção do conjunto de treinamento. Quando fazemos $h = 1..M$ em $\vec{\xi}^{\mu\zeta}$ temos M vetores, cada vetor ser visto como um ponto numa curva embebida no espaço de N dimensões e parametrizada pela equação de $S(q, H)$; Como cada uma das concentrações entre H_h é escolhida aleatoriamente no intervalo $H_{min}.....H_{max}$, com distribuição uniforme; esperamos que os intervalos $H_{\zeta}^{\mu} - H_{\zeta+1}^{\mu}$ entre pontos sejam da ordem de:

$$\delta H = \frac{H_{max} - H_{min}}{M}. \quad (4.11)$$

A pergunta que pretendemos responder é:

De que ordem deve ser δH de modo que possamos esperar que os resultados fiquem independentes de M ?

Note que um δH muito pequeno resultaria num tempo de treinamento muito longo, enquanto que um δH muito grande faria com que a baixa definição das curvas, não garanta que elas sejam realmente separadas.

Chamemos de conjuntos de treinamento equivalentes aqueles que têm o mesmo número de exemplos p , para diferentes números de concentrações por odor M , e escolhamos uma arquitetura para o percéptron.

¹ $H_{min} = 0$ significa que não existe odorante; não se pode clasificar odorantes que não existem.

Esperamos que conjuntos de treinamento equivalentes tenham probabilidade iguais de serem aprendidos em percéptron de arquitetura idêntica.

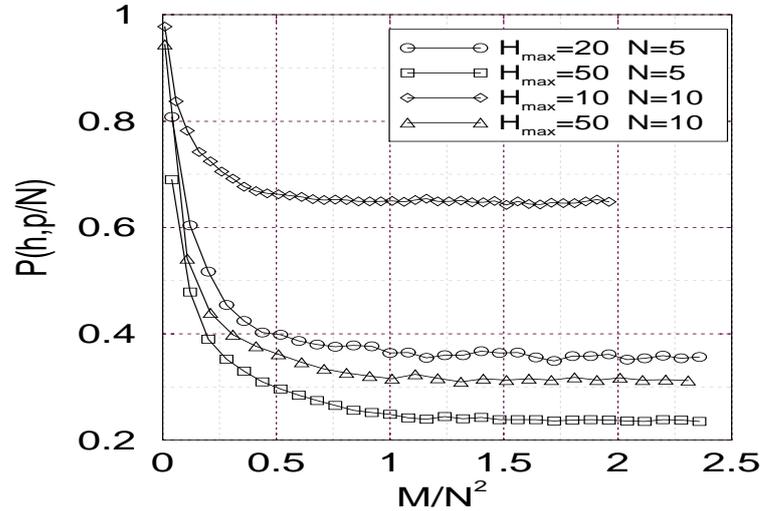


Figura 4.1: Probabilidade de um conjunto de treinamento ser “aprendido” vs. M/N^2 , onde fixamos o número de odorantes $p = 25$.

A figura (4.1) mostra os resultados do processo de treinamento, com uma arquitetura dada e o número de odorantes constante $p = 25$; o número máximo de concentrações por odorantes é o único parâmetro que muda. Se o aprendizado deve ser independente do número de concentrações por odor, para estas características do percéptron e do conjunto de treinamento, espera-se que as curvas $P(h, \frac{p}{N})$ sejam constantes (linhas retas). A figura 4.1 mostra que isto não acontece para pequenos valores de M . No entanto, notamos que se M é ao menos igual ao quadrado do número de entradas do percéptron, a probabilidade pode-se considerar independente do número de concentrações por odorante.

Dada tal evidência experimental, assumiremos neste trabalho $M = N^2$.

4.2 Construindo as curvas $P(h,p/N)$ vs. p/N

Uma maneira de se computar experimentalmente o alpha crítico é construindo as curvas de probabilidade de que um conjunto seja aprendido vs. p/N , onde p é o número de exemplos. Neste caso, p é o número de odores e N a dimensão do percéptron. Primeiro, vamos definir o que é um conjunto de treinamento aprendível.

1. **Conjunto aprendível:** é aprendível se é linearmente separável, No contexto do algoritmo, o conjunto é aprendível se a condição $\vec{w} \cdot \xi^{\mu\zeta} > 0$ é satisfeita para todo memebro do conjunto.
2. **Conjunto não aprendível:** é não aprendível se é linearmente não separável. No contexto do algoritmo, é não aprendível o conjunto se no processo de treinamento a condição $d > d_c$ é satisfeita.

A probabilidade de que um conjunto de treinamento seja aprendível para um H_{max} e computada como:

1. Escolhemos a arquitetura do percéptron. Neste caso, a escolha de N determina sua arquitetura.
2. Geramos p vetores de afinidade de componentes q_i e de dimensão N . As componentes q_i são computadas de acordo com a distribuição $\sigma(q)$ de média $\langle \sigma(q) \rangle = 1$ e variância $\sigma = 1$
3. Por cada odorante são gerados M [$H_{min} = 0.5, H_{max}$] vetores de concentração, onde $M = N^2$, de modo que finalmente tenhamos um conjunto de treinamento composto por $p \cdot M$ vetores, de dimensão N .
4. Os vetores deste conjunto de treinamento serão apresentados ao percéptron aleatoriamente, até que uma das duas condições de aprendizado seja satisfeita.
5. Deste modo são gerados Φ conjuntos de treinamento, é a probabilidade e computada como: $P(h, \frac{p}{N}) = \frac{\text{Numero de conjuntos de treinamento aprendidos}}{\text{Numero de conjuntos de treinamento apresentados}}$.
6. Os passos de 1 a 5 se repetem trocando cada vez o valor de p .

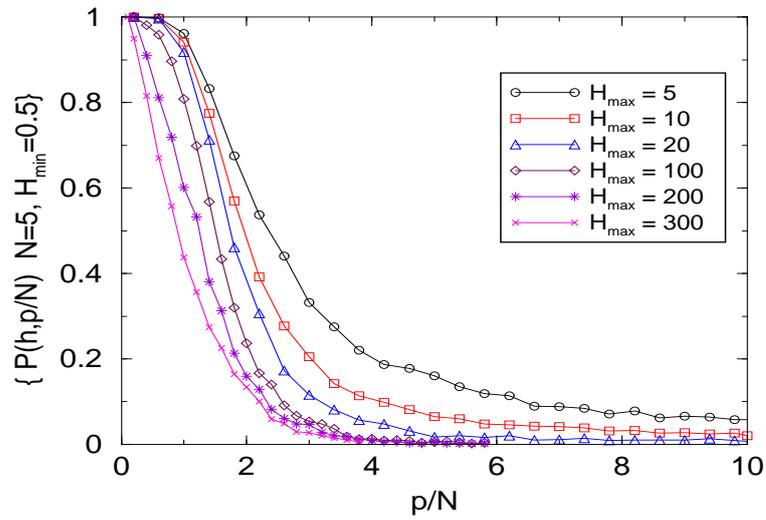


Figura 4.2-A: Probabilidade vs. p/N . Cada curva corresponde a uma concentração diferente. O número de entradas do percéptron é $N = 5$.

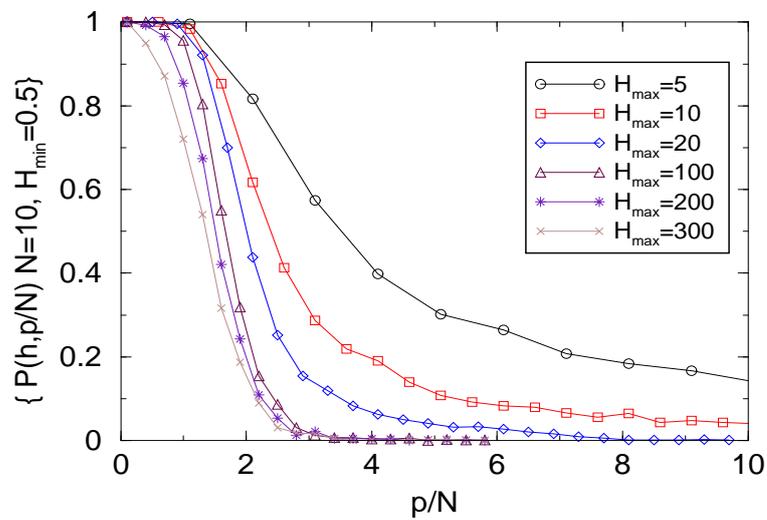


Figura 4.2-B: Probabilidade vs. p/N . Cada curva corresponde a uma concentração diferente. O número de entradas do percéptron é $N = 10$.

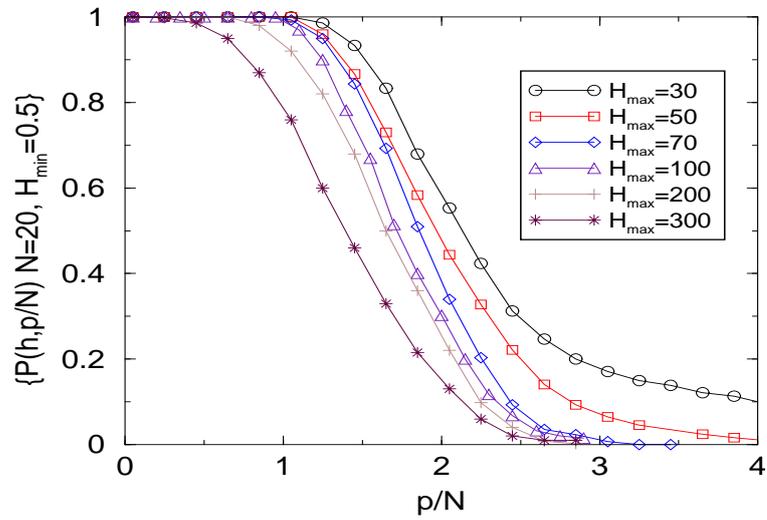


Figura 4.2-C: Probabilidade vs. p/N . Cada curva corresponde a uma concentração diferente. O número de entradas do percéptron é $N = 20$.

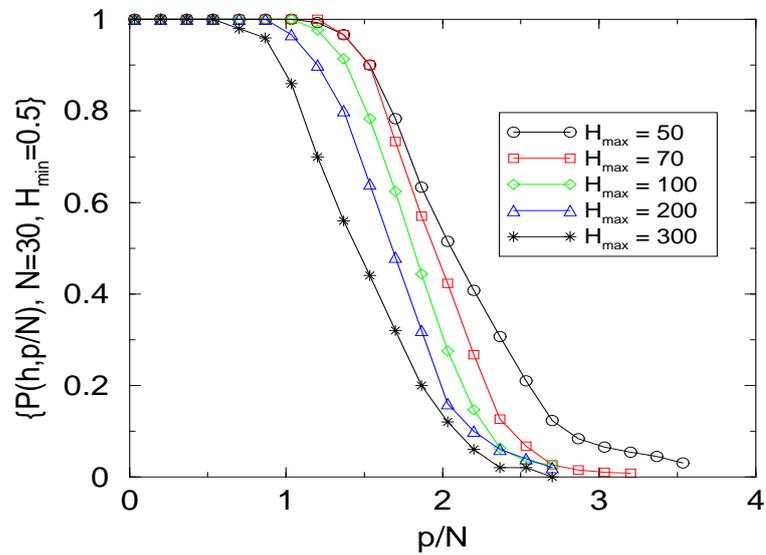


Figura 4.2-D: Probabilidade vs. p/N . Cada curva corresponde a uma concentração diferente. O número de entradas do percéptron é $N = 30$.

As figuras 4.2-A, 4.2-B, 4.2-C, 4.2-D mostram as curvas de probabilidade para percéptrons de 5, 10, 20 e 30 entradas respectivamente. Cada uma das curvas nas figuras, corespondem a um valor de H_{max} . Cada ponto nas curvas é gerado ao se fazer a média aritmética de 100 realizações. Isto é, para cada ponto nas curvas são gerados e apresentados ao percéptron 100 conjuntos de treinamento. Por exemplo, se todos os 100 foram aprendidos, $P = 1$, se 10 foram aprendidos, $P = 10/100$, etc.

4.3 Capacidade crítico no problema de odorantes α_c

Definimos o parâmetro α da seguinte forma: $\alpha = \frac{p}{N}$, que chamamos de carga do percéptron, onde p é o número de odorantes e N é o número de entradas do percéptron. Dado que só faz sentido falar de capacidade máxima no limite termodinâmico, definimos um alpha crítico α_c , que chamamos de capacidade crítica do percéptron (ou carga máxima do percéptron) no limite de N grande, definido da seguinte forma: $\alpha_c = \lim_{N \rightarrow \infty} \frac{p}{N}$. Dado que as simulações são restringidas a $N < 30$, é preciso estabelecer um método que nos permita determinar α_c no limite $N \rightarrow \infty$.

4.3.1 Análise FFS

O método de tratamento de dados “Finite Size Scaling” (FSS)[15] provou ser uma ferramenta fundamental na análise de transições de fase, as quais ocorrem rigorosamente no limite termodinâmico. FSS é usado em simulações de sistemas de tamanhos finitos.

Transições de fase são comuns em percolação e matéria condensada, no entanto, elas também ocorrem em gráficos aleatórios, RNs e algoritmos de busca. Kirkpatrick e Selman [16] mostraram que FSS pode ser eficiente em problemas sim escala de comprimento intrínseca. RNs são sistemas sem escala de comprimento intrínseca. Nesta dissertação proporemos um análises FSS como método de tratamento de dados, para determinar a capacidade no limite termodinâmico.

Nas figuras 4.2-A, 4.2-B, 4.2-C e 4.2-D é possível perceber que a queda das curvas $P(h,p/N)$ se faz mais acentuada para valores de N mais grades (note o cambio da escala nas figuras). Para valores iguais de H_{min} e H_{max} , curvas para diferentes valores de N se cruzam aproximadamente no mesmo valor de de $\alpha = p/N$ (figuras

4.3-A ... 4.9-A), isto é análogo a resultado de Cover de parâmetros aleatórios. Observe no entanto, que o cruzamento não é para $P(h, p/N) = 1/2$ mas para um valor diferente (menor). Um fenômeno similar com um cruzamento o cima da probabilidade $1/2$ é observado em outras arquiteturas, como em maquina comitê e maquina paridade[17]. Esperamos que no limite termodinâmico as curvas mostrem uma transição aguda, a qual poderia indicar que $\alpha_c(H_{min}, H_{max})$, onde um conjunto de treinamento serão aprendido com probabilidade 1 se $\alpha < \alpha_c(H_{min}, H_{max})$ e um conjunto serão aprendido com probabilidade 0 se $\alpha > \alpha_c(H_{min}, H_{max})$. Mesmo na ausência de um comprimento de correlação típica, é razoavel abordar o problema com um análise FSS.

Neste trabalho proporemos uma simples reescala da variável α , com 2 parâmetros

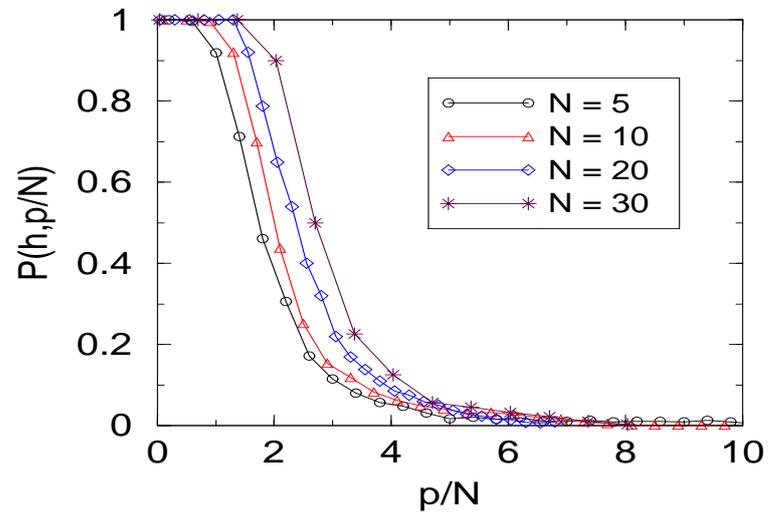
$$y = (\alpha - \tilde{\alpha}_c)N^{\frac{1}{\nu}}. \quad (4.12)$$

As figuras 4.3-B...4.9-B mostram a reescada de dados. O colapso das curvas permite concluir que o sistema apresenta uma transição de primeira orden em $\alpha_c(H_{min}, H_{max})$.

4.3.2 Simulações

As figuras 4.3-A ... 4.9-A mostram as curvas de probabilidade vs. p/N . Em cada gráfico são traçadas 4 curvas; cada curva corresponde a um percéptron de arquitetura diferente. A dimensão do cada percéptron é $N=5$, $N=10$, $N=20$ e $N=30$. Note-se que as curvas se interceptam aproximadamente no mesmo valor de $\frac{p}{N}$, alpha critico deve se encontrar precisamente no ponto de intersecção das curvas. As figuras 4.3-B ... 4.9-B, mostram as curvas reescaladas, de acordo com a equação (4.12). As curvas nas figuras x-A não se cruzam exatamente no mesmo ponto, de modo que temos certa liberdade de escolha em α_c , e o parâmetro ν foi escolhido de modo que o colapso seja o melhor possível. A análise de dados FSS e o bom colapso das curvas nos permite garantir que o α_c encontrado em cada caso corresponde à capacidade do percéptron no limite termodinâmico $N \rightarrow \infty$.

A



B

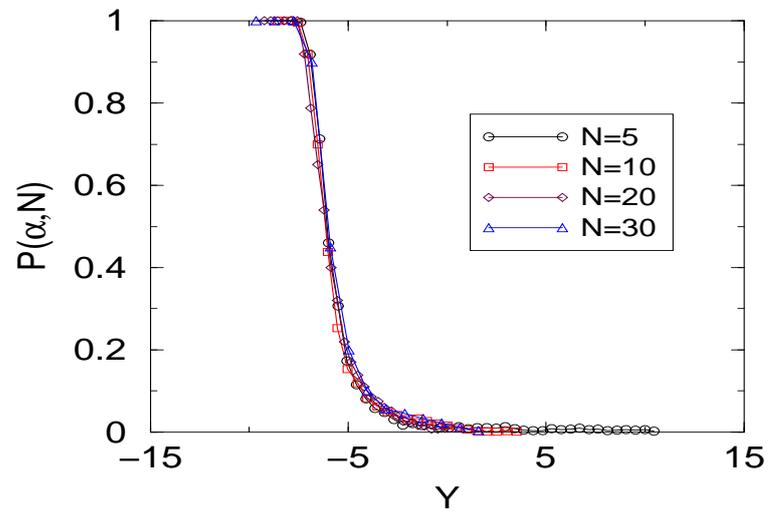
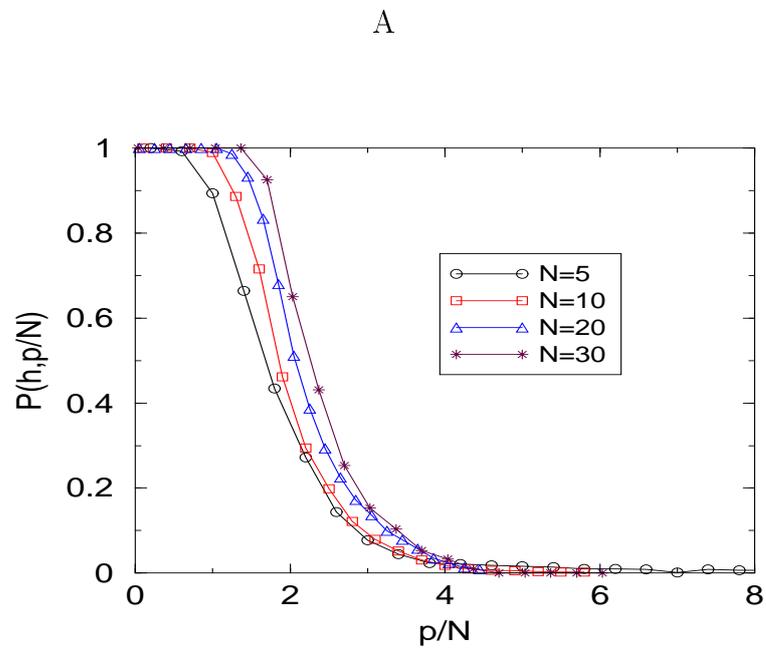


Figura 4.3-A Probabilidade de aprendizado vs. p/N para $H_{max} = 20$ para o percéptron de 5, 10, 20 e 30 entradas. O cruzamento das curvas de probabilidade ocorre aproximadamente em: $\alpha_c = 6.9$. As curvas de 4.3-B foram reescaladas de acordo com a equação (4.12), onde $\nu = 10.0$.



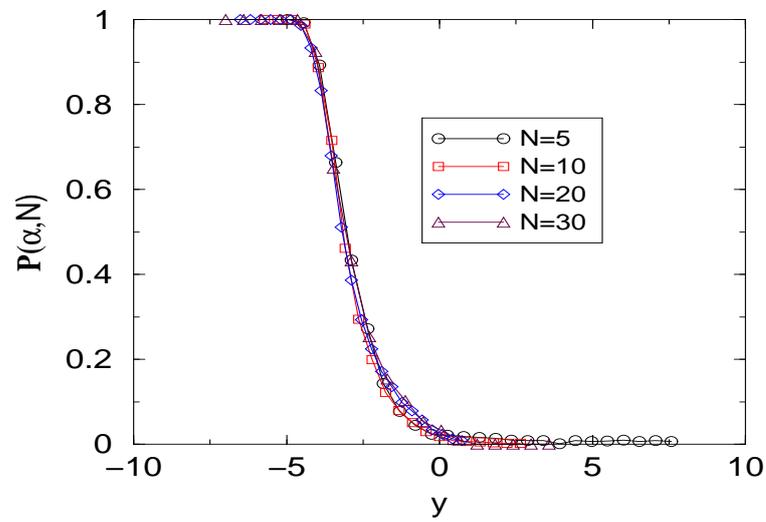


Figura 4.4-A Probabilidade de aprendizado vs. p/N para $H_{max} = 30$, para o perceptron de 5, 10, 20 e 30 entradas. O cruzamento das curvas das curvas probabilidade ocorre aproximadamente em: $\alpha_c = 4.0$. As curvas de 4.4-B foram reescaladas de acordo com a equação (4.12), onde $\nu = 6.0$.

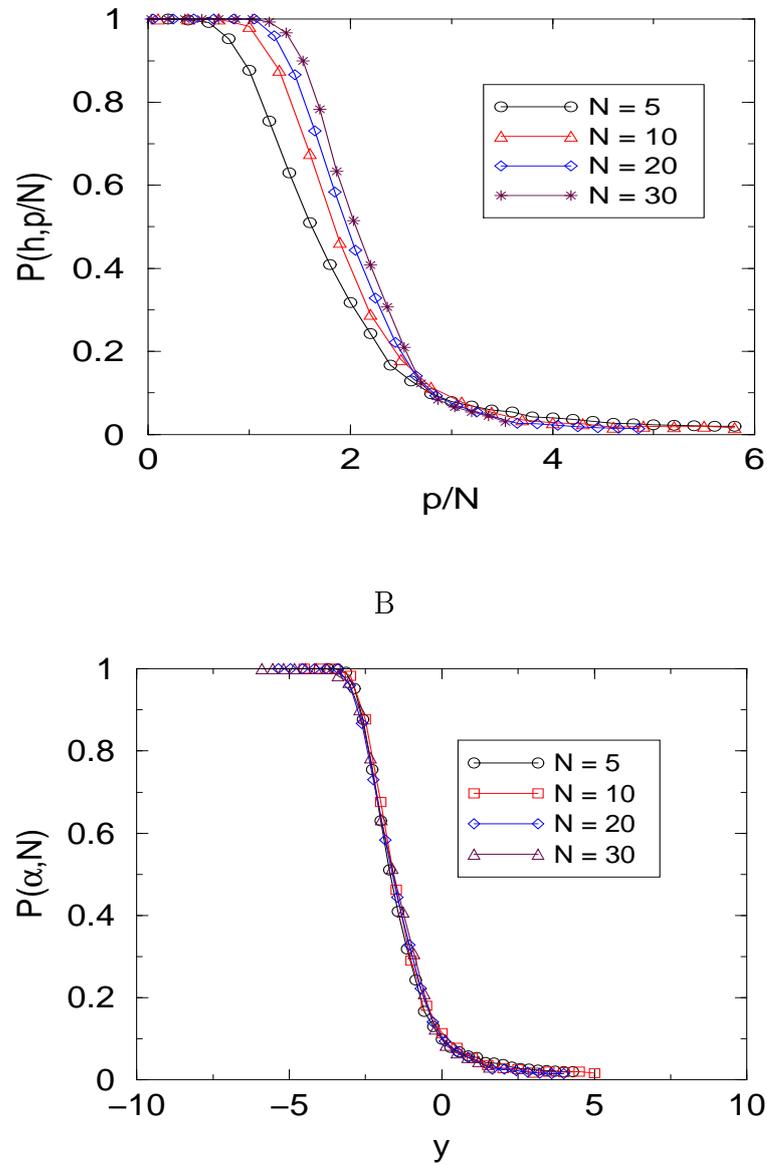


Figura 4.5-A Probabilidade de aprendizado vs. p/N para $H_{max} = 50$, para os percéptrons de 5, 10, 20 e 30 entradas. O cruzamento das curvas de probabilidade ocorre aproximadamente em: $\alpha_c = 2.8$. As curvas de 4.5-B foram reescaladas de acordo com a equação (4.12), onde $\nu = 4.5$

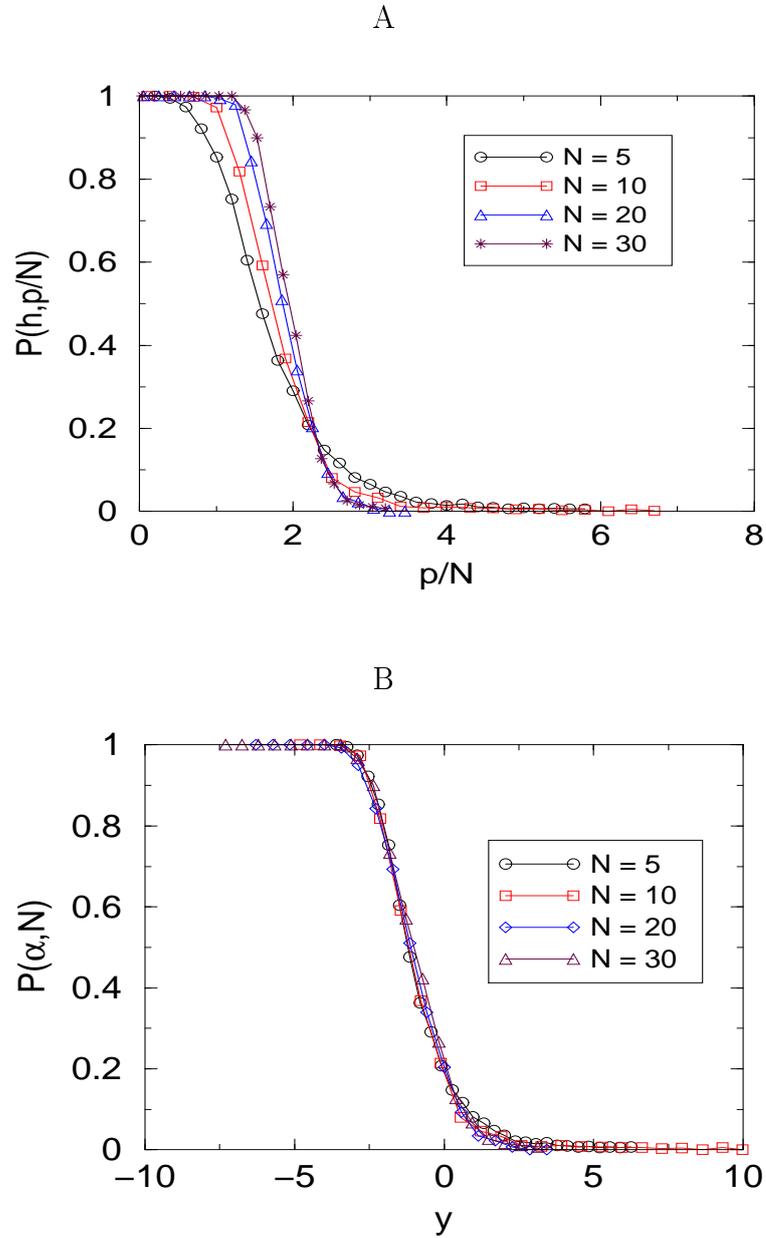


Figura 4.6-A: Probabilidade vs. p/N para $H_{max} = 70$, para os percéptron de 5, 10 20 e 30 entradas. O cruzamento das curvas de probabilidade ocorre aproximadamente em: $\alpha_c = 2.3$. As curvas de 4.6-B foram rescaladas de acordo com a equação (4.12), onde $\nu = 2.85$

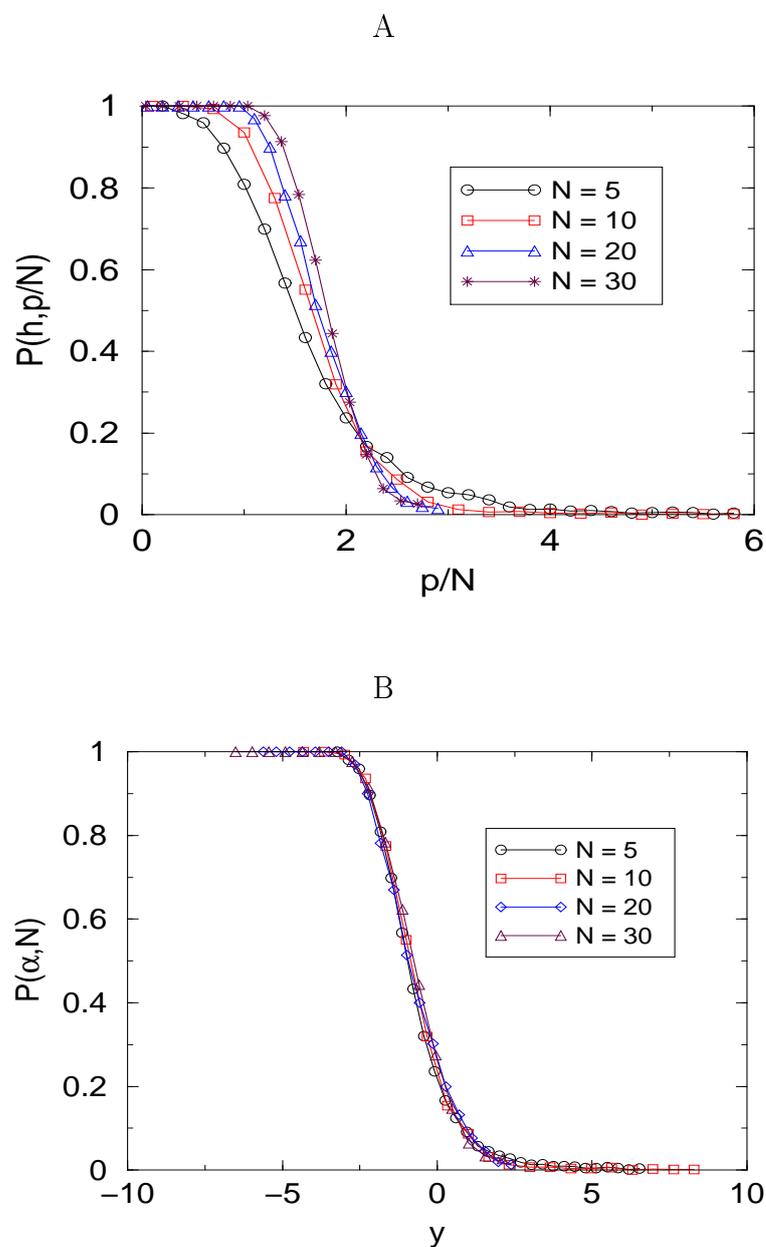


Figura 4.7-A: Probabilidade vs. p/N para $H_{max} = 100$, para os percéptron de 5, 10, 20, e 30 entradas. O cruzamento das curvas de probabilidade ocorre aproximadamente em: $\alpha_c = 2.2$. As curvas de 4.7-B foram reescaladas de acordo com a equação (4.12), onde $\nu = 2.75$

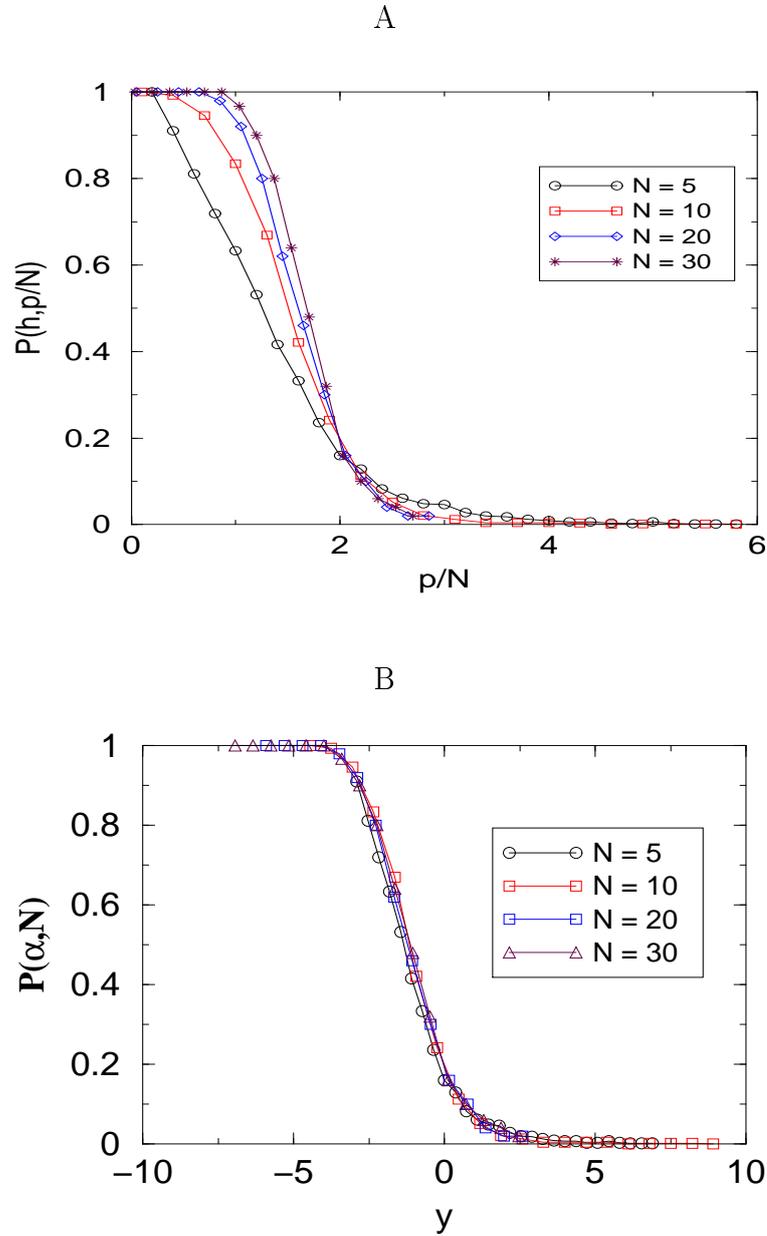


Figura 4.8-A: Probabilidade vs. p/N para $H_{max} = 200$, para os percéptron de 5, 10, 20, e 30 entradas. O cruzamento das curvas de probabilidade ocorre aproximadamente em: $\alpha_c = 2.1$. As curvas de 4.8-B foram reescaladas de acordo com a equação (4.12), onde $\nu = 2.7$

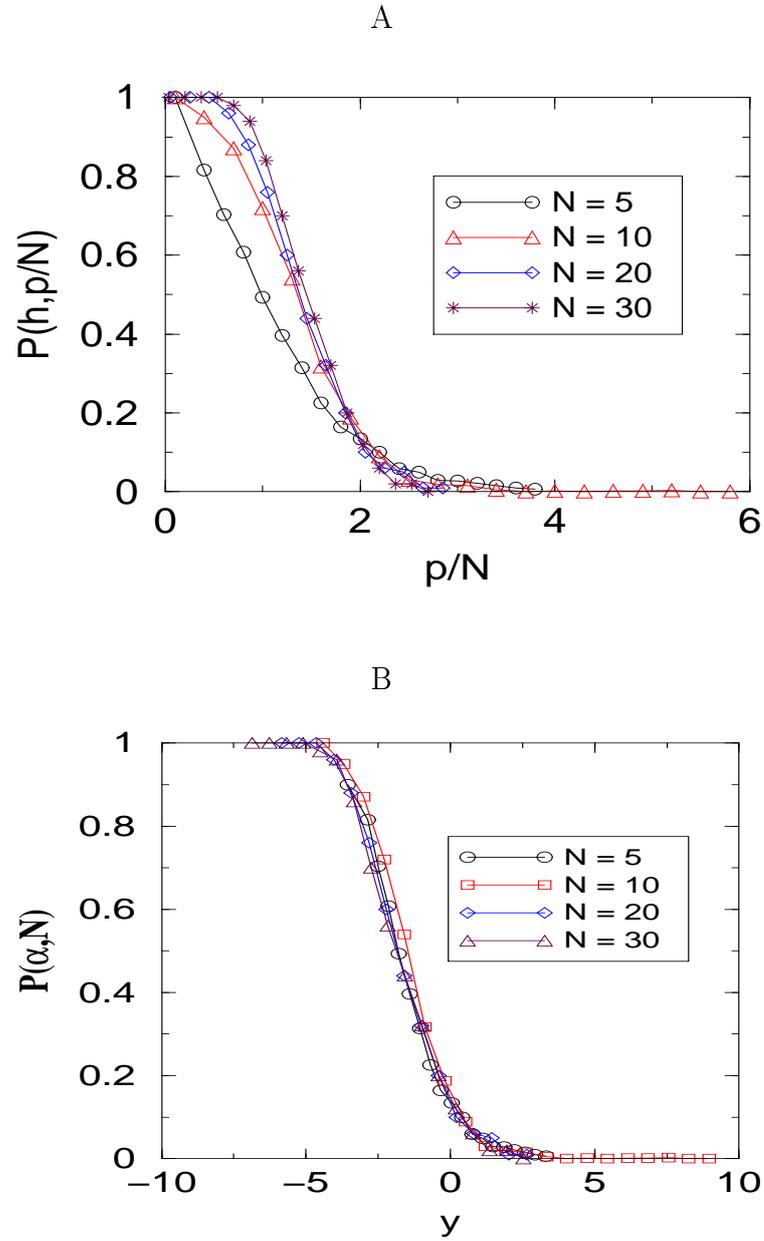


Figura 4.9-A: Probabilidade vs. p/N para $H_{max} = 300$, para o percéptrons de 5, 10, 20 e 30 entradas. O cruzamento das curvas de probabilidade ocorre aproximadamente em: $\alpha_c = 2.1$. As curvas de 4.9-B foram reescaladas de acordo com a equação (4.12), onde $\nu = 2.7$

H_{max}	α_c	ν
5	21.0	15.0
10	12.0	12.0
20	6.90	10.0
30	4.00	6.00
50	2.80	4.50
70	2.25	2.85
100	2.15	2.75
200	2.00	2.70
300	2.00	2.70

Tabela 4.1: Valores de H_{max} , α_c e ν extraídos dos pontos de intersecção nas figuras 4.3 ... 4.9.

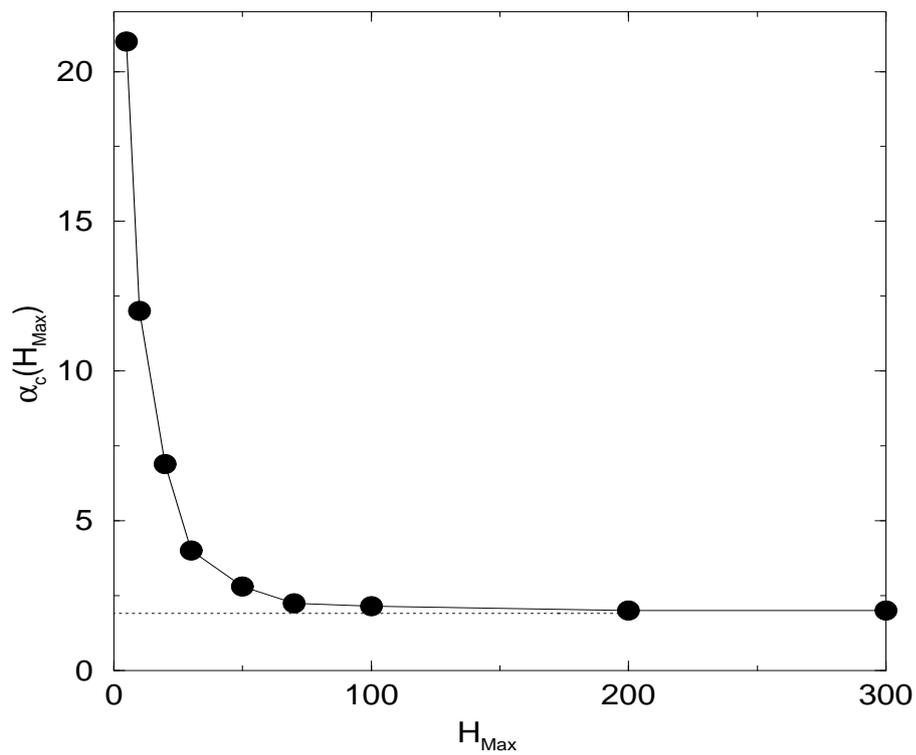


Figura 4.12: Capacidade crítica vs. concentração máxima. Para altas concentrações, a capacidade do percéptron vai para 2.

A figura 4.12 mostra α_c vs. H_{max} . Cada ponto na curva corresponde a α_c nas curvas das figuras 4.3 .. 4.9. No limite de altas concentrações, a capacidade crítica vai para 2. É interessante notar que a capacidade crítica do percéptron simples para padrões aleatórios é precisamente 2.

Figura 4.13: O parametro ν vs. concentração máxima. Para altas concentrações, ν vai para 2.7.

Capítulo 5

Conclusões

5.1 Interpretação geométrica

Separar pontos de 2 espécies diferentes num espaço de N dimensões é um problema largamente estudado. RNs podem encontrar planos que separem o espaço em duas regiões, de modo que pontos de espécies iguais fiquem numa mesma região do espaço. O problema proposto neste trabalho pode ser visto como sendo a separação de 2 espécies diferentes de linhas por hiperplanos num espaço de N dimensões. Interpretamos o problema de odorantes como:

Para cada odor μ temos M vetores; cada vetor pode-se entender como um ponto num espaço de N dimensões, de coordenadas $\{S_1^{\mu\zeta} \dots S_N^{\mu\zeta}\}$; todos esse pontos desenham uma curva no espaço, que corresponde ao odor μ , parametrizado pela concentração.

Devido a $\mu = 1 \dots p$, teremos p curvas embebidas neste espaço, cada uma delas correspondendo a um odor diferente. Essas curvas podem ser melhor entendidas fazendo projeções dos vetores no plano $(S_1^{\mu\zeta}, \vec{w} \cdot \vec{S}^{\mu\zeta})$, e dado que \vec{w}^* pode ser interpretado como sendo perpendicular ao plano que separa a curva do odorante correto dos demais, se olharmos na direção de \vec{w}^* veremos a curva que corresponde a odor que queremos classificar separada das demais. Numa direção qualquer \vec{w} , todas as curvas ficam confundidas.

Figura 5.1: Projeção das curvas de odor no plano $\vec{w} \cdot \vec{X}^{\mu\zeta}$. Nos gráficos, cada curva corresponde a um odor e cada ponto na curva corresponde a uma concentração diferente, onde $N = 20$ $H_{max} = 30$ $p = 45$ $M = 400$. As curvas em [A] foram projetadas numa direção qualquer, enquanto que as curvas em [B] foram projetadas na direção do vetor \vec{w}^* . A curva de traço contínuo corresponde ao odor que pretendemos separar.

As figuras 5.1, 5.2, e 5.3 mostram a projeção das curvas de odorantes no plano $(\vec{w} \cdot \vec{X}^{\mu\zeta})$ onde $\vec{X}^{\mu\zeta} = \frac{\vec{S}^{\mu\zeta}}{\sqrt{\sum_{i=1}^N (S_i^{\mu\zeta})^2}}$. Cada curva nas figuras corresponde a um odor.

Uma curva no espaço de N dimensões é desenhada pelos vetores $\{\vec{X}^{\mu 1} \dots \vec{X}^{\mu M}\}$, de modo que quando $\mu = 1..p$ temos p curvas desenhadas neste espaço. Aprender no contexto de odorantes significa separar a curva desenhada pelos vetores $\{\vec{X}^{11} \dots \vec{X}^{1M}\}$ das curvas $\{\vec{X}^{\mu 1} \dots \vec{X}^{\mu M}\}$ $\mu = 2..p$; Se é possível separar as curvas, o problema é linearmente separável.

Na figura 5.1-A, mostramos o conjunto de treinamento de parâmetros: $N = 20$ $H_{max} = 30$ $p = 45$ $M = 400$. As curvas foram projetadas no plano $(\vec{X}^{11} \cdot \vec{X}^{\mu \zeta})$, onde as curvas ficam todas confundidas. A curva que corresponde ao odor que queremos separar está marcada com uma seta. Na figura 5.1-B mostramos o mesmo conjunto de curvas que em 5.1-A depois de executar o algoritmo de treinamento e encontrar o vetor \vec{w}^* , que satisfaz a condição $\vec{w} \cdot \xi^{\mu \zeta} > 0$, projetamos as curvas no plano $\vec{w}^* \cdot \vec{X}^{\mu \zeta}$, as curvas foram classificadas corretamente (o problema é linearmente separável), portanto o vetor \vec{w}^* é perpendicular ao plano de separação das curvas.

A figura 5.2 mostra o mesmo conjunto de curvas que 5.1 depois de se mudar o conjunto de treinamento de $\mathcal{L}\{N = 20, H_{max} = 30, p = 45 M = 400\}$ para $\mathcal{L}\{N = 20, H_{max} = 300, p = 45 M = 400\}$. Note que a concentração máxima foi de $H_{max} = 30$ para $H_{max} = 300$. Agora, muitas das curvas cruzam o plano de separação (figura 5.2-B) encontrado em 5.1-B. Resta assim tratar de aprender um novo plano que possa separar este novo conjunto de treinamento. A figura 5.3 mostra a mesma figura 5.2-B depois de executado o algoritmo de treinamento. A curva que pretendemos separar está marcada por uma seta; neste caso não é possível traçar um plano que separe o conjunto, o que nos permite concluir que o problema é linearmente não separável.

Figura 5.2: Mostra as mesmas curvas que 5.1-A, e 5.1-B projetadas no plano $\vec{w}^* \cdot \vec{X}^{\mu\zeta}$, depois de modificado o conjunto de treinamento. $N = 20$ $H_{max} = 300$ $p = 45$ $M = 400$. A figura [B] é uma ampliação de [A]. Os vetores \vec{w}^* em 5.1-B e 5.2 são idênticos.

Figura 5.3: Mostra uma ampliação das curvas em 5.1, projetadas na direção obtida após de tentar aprender o conjunto modificado em 5.2. Neste caso, a mudança nos valores de concentração máxima fazem com que um problema linearmente separável fique linearmente não separável.

5.2 Conclusões

Estudamos computacionalmente um modelo probabilista de odores baseado na distribuição de afinidades e concentrações, com o objetivo principal de calcular o α_c nesse modelo. O cálculo de α_c pretende contribuir na compreensão do mecanismo de percepção de odores, embora na sua relevância biológica possa estar seu major mérito, o trabalho também contribui na apresentação de um algoritmo claro e geral para o cálculo numérico de α_c .

Os resultados podem ser resumidos basicamente nas seguintes assertivas:

1. **A capacidade crítica é uma função da concentração máxima** $\alpha_c = \alpha(H_{max})$. Isto é uma consequência natural do modelo, dado que a altas concentrações os transdutores se saturam (a concentração em que um transdutor entrará no regime de saturação depende do grau de afinidade do transdutor para o odor em questão). À medida em que a concentração máxima aumenta,

a resposta dos transdutores ficam todas perto de 1, isto faz com que a capacidade do percéptron de reconhecer odores fique comprometida, isto é, diminui com o aumento da concentração máxima.

2. **Para altas concentrações, a capacidade crítica vai para 2.** $\alpha_c(H_{max}) \rightarrow 2$ se $H_{max} \gg 1$. A capacidade crítica de um percéptron para padrões completamente descorrelacionados é 2, e dado que o modelo estudado nesta dissertação exibe um alto grau de correlação, ao menos para pequenas concentrações, a capacidade crítica deve ser maior que 2.

A razão pela qual $\alpha_c(H_{max} \gg 1) \rightarrow 2$ está em que a escolha de uma concentração é aleatória com distribuição contínua entre H_{min} e H_{max} .

Lembre-se que, na seção 4.1.1, provamos que se M é pelo menos igual que o quadrado do número de entradas do percéptron, a capacidade crítica é considerada independente do número de concentrações M por odorante. Aumentos na concentração significan *aumentar a densidade de pontos e o comprimento das curvas de odores nos extremos de alta concentração*.

Aumentos na densidade de pontos não são relevantes dado que a definição das curvas está garantida desde que $M = N^2$. Qualquer aumento na densidade acima de N^2 não modifica os resultados; assim, aumentos na concentração são só relevantes para o problema na medida em que aumentam o comprimento das curvas nos extremos de alta concentração.

Esses aumentos no comprimento das curvas fazem com que as curvas de odorantes fiquem cada vez mais próximas em seus extremos de alta concentração. É claro que a capacidade do percéptron para separar curvas cada vez mais próximas deve decair.

Se impusermos a condição de que as curvas adotem o valor de concentração máxima, dependendo da afinidade dos transdutores e H_{max} , os transdutores ficarão em regime de saturação. Isto fará com que todas as curvas de odores tenham um ponto comum, de modo que a capacidade $\alpha_c(H_{max} = \infty) = 0$.

Para responder a pergunta: “porque α_c tende a um valor fixo para $H_{max} \gg 1$?” construímos a figura 5.4-A, onde graficamos os pontos dos extremos de concentração máxima nas curvas de odor Vs. H_{max} . Notemos também que as curvas de odorantes são lineares em seus extremos de alta concentração (figura 5.4-B)

e se aceitamos o fato de que α_c depende de que tão próximas estejam as curvas em seus extremos de alta concentração, é óbvio ao se analisar a gráfica que α_c deve tender a um valor fixo pelo menos para $Hmax > 200$.

A

B

Figura 5.4: O gráfico 5.4-A mostra quão próximos estão os transdutores da saturação à medida que a concentração máxima do odorante aumenta. O gráfica 5.4-B mostra a linearidade das curvas de odor, Nos extremos de alta concentração.

1. É importante também observar que:
 - O método estudado nesta dissertação para calcular o α_c , baseado no cálculo das curvas de estado $P(h, p/N)$ vs. p/N e na aplicação do tratamento de dados FSS, é um método importante. Ele nos permite calcular numericamente α_c em qualquer situação, sempre que o problema seja computacionalmente acessível.
 - O problema de se separar curvas em espaços de N dimensões, por hiperplanos, é um problema matemático interessante. Não temos conhecimento de trabalhos prévios nesse tópico. No entanto, há muitos trabalhos sobre separação de pontos em espaços de N dimensões.

Embora o trabalho[19] seja importante como um algoritmo numérico para o cálculo de α_c , o trabalho não pode ser completado sem uma interpretação cuidadosa da importância dos resultado no mecanismo de percepção de odores biológicos.

Referências Bibliográficas

- [1] McCulloch, W.S., Pitts, W. (1943). -A logical calculus of the ideas immanent in nervous activity. Bull. Math. Biophys. 5, 115-133
- [2] Hebb, D.O. (1949). -The Organization of Behavior. New York: Wiley. Partially reprinted in Anderson and Rosenfeld[1988]
- [3] Rosenblatt, F. (1962).- Principles of Neurodynamics. New York: Spartan.
- [4] Minsky, M.L. and S.A. Papert (1969).- Perceptrons. Cambridge: MIT Press. Partially reprinted en Anderson and Rosenfeld[1988]
- [5] Rumelhart, D.E., G.E. Hinton, and R. J. Williams (1986).- Learning Representations by Back-propagating errors. Nature 323, 533-536. reprinted en Anderson and Rosenfeld[1988]
- [6] Hopfield, J.J. (1987). Learning Algorithms and Probabilty Distributions in Feed_Forward and Feed_Back Networks. Proceedings of the National academy of Sciences, USA 84, 8429-8433.
- [7] Little, W.A. (1974). The Existence of Persistent States in the Brain. Mathematical Biosciences 19, 101-120
- [8] Gardner, E. and B. Derrida (1988).- The Space of Interactions in Neural Network Models. Journal of Physics A 21, 257-270.
- [9] The Journal of Neuroscience Smell and the Olfactory System 1995 S. Camper, F. Probst, Y. Raphael.
- [10] D. Lancet, E. Sadovsky and E. Seidman, Proc. Natl. Acad. Sci. USA.

-
- [11] D. Nabutovsky and E. Domany, (1991).- *Neural Computation*, 3, 604.
- [12] *The Theory of Matrices Second Editions* Peter Lancaster and Miron Tismenetsky.
- [13] Cover, T.M. (1965).- *Geometrical and Statistical Properties of Systems of Linear Inequalities with Applications in Pattern Recognition*. *IEEE Transactions on Electronic Computers* 14, 326-334.
- [14] Gardner, E. (1988).- *Maximum Storage Capacity in Neural Networks*. *Euro Physics Letters* 4, 481-485.
- [15] V. Privman (1990).- *Finite Size Scaling and Numerical Simulations of Statistical Systems* World Scientific, Singapore.
- [16] S. Kirkpatrick and B. Selman, (1994).- *Science* 264, 1297.
- [17] Walter Nadler and Wolfgang Fink .- *Finite Size Scaling in Neural Networks*. *Physical Review Letters* 78, 555-558 Number 3 1997
- [18] Craig A. Lindley 1991.- *Practical Image Processing in C* Wiley Professional computing.
- [19] Trabalho a ser publicado Tejada, Lancet, Domany, Caticha. Em preparação.