Perceptrons

Renato Vicente

3 de maio de 2017

1 Neurônio de McCulloch-Pitts

Warren S. McCulloch e Walter H. Pitts foram dois pioneiros do aprendizado de máquinas.McCulloch (1898-1969) estudou filosofia e psicologia em Yale e se tornou médico psiquiatra em 1927 na Columbia. Em 1942 McCulloch era professor no departamento de psiquiatria da Universidade de Chicago quando conheceu Pitts (1923-1969), um jovem de 19 anos extremamente brilhante e sem-teto. McCulloch então convidou Pitts para morar com sua família e os dois passaram a colaborar. Pitts era autodidata. Aos 12 anos de idade leu *Principia Mathematica* de Whitehead e Russell em três dias e enviou uma carta a Bertrand Russell apontando problemas no primeiro volume. Russell respondeu a carta convidando-o para estudar no Reino Unido, a partir disso o garoto passou a se dedicr à lógica e a matemática.

Em 1943, Warren S. McCulloch e Walter H. Pitts publicaram um artigo intitulado *A Logical Calculus of the Ideas Immanent in Nervous Activity* propondo que neurônios implementariam operações lógicas. Fizeram isso simplificando neurônios biológicos na forma de um modelo representado na figura 1.

Após o trabalho de Rosenblatt em 1962 os neurônios de McCulloch-Pitts passaram a ser denominados *Perceptrons*. Um Perceptron implementa a classe de hipóteses composta por funções afins:

$$H := \{\varphi(\boldsymbol{W}\cdot\boldsymbol{x}+W_0) : \boldsymbol{W},\boldsymbol{x}\in\mathbb{R}^n; W_0\in\mathbb{R}\}$$

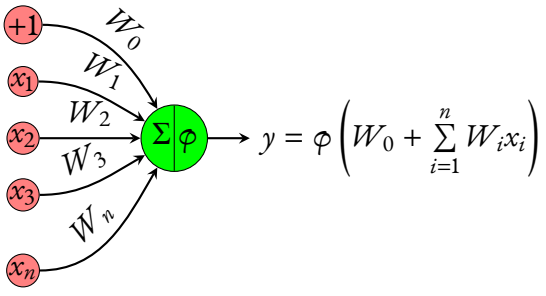

Figura 1: Neurônio de McCulloch-Pitts

Escolhas populares para a *função de transferência* $\phi(u)$ são:

1. Perceptron Booleano: $\phi(u) = \text{sign}(u)$ ou $\phi(u) = \Theta(u)$, onde

$$\Theta(u) = \begin{cases} 1 & \text{se } u \leq 0 \\ 0 & \text{caso contrário,} \end{cases}$$

   é a função de Heaviside.

2. Perceptron Linear: $\phi(u) = u$.

3. Perceptron Logístico: $\phi(u) = \tanh(u)$ ou

$$\phi(u) = \frac{1}{1 + e^{-u}}$$

4. Perceptron ReLU: $\phi(u) = \max(0, u)$.

2 Perceptron Booleano e Separabilidade Linear

Definição 1. *Um conjunto de treinamento* $S = \{(\boldsymbol{x}_\mu, y_\mu) : \mu \in [m], \boldsymbol{x}_\mu \in R^n, y_\mu \in \{\pm 1\}\}$ *é linearmente separável se existe um* $\boldsymbol{W} \in \mathbb{R}^{n+1}$ *tal que* $(\boldsymbol{W} \cdot (1, \boldsymbol{x}_\mu))y_\mu > 0$ *para todo* $\mu \in [m]$.

Por simplicidade, deste ponto em diante, trataremos apenas do caso $H := \{\varphi(\boldsymbol{W} \cdot \boldsymbol{x}) : \boldsymbol{W}, \boldsymbol{x} \in \mathbb{R}^n\}$.

Como consequência, se a regra é gerada por um Perceptron Booleano sem ruído, então ela é linearmente separável. Em 1969, Marvin Minsky (1927-2016) e Seymor Papert (1928-2016) publicaram o livro *Perceptrons: an introduction to computational geometry*. Este livro trás uma série de resultados matemáticos sobre Perceptrons, em particular, ressalta a incapacidade de se implementar em um neurônio simples a operação lógica XOR, que não é linearmente separável. O livro também demonstra que mesmo redes com uma camada escondida são incapazes de implementar certas operações lógicas a menos que pelo menos um de seus neurônios da camada interna esteja conectado com todas as entradas (contrariando a intuição da época de que era necessário que localidade dos neurônios seria uma propriedade desejável).

Para um dado conjunto de m de vetores $\{\boldsymbol{x}_\mu : \boldsymbol{x}_\mu \in \mathbb{R}^n, \mu \in [m]\}$ é possível construir 2^m diferentes conjuntos de treinamento S. Uma pergunta imediata é: quantos destes conjuntos de treinamento seriam linearmente separáveis?

3 Capacidade de Representação: Teorema de Contagem de Funções de Cover

Definição 2. *Em um espaço afim de n dimensões, um conjunto de pontos está em posição geral se não mais do que* $k + 1$ *destes pontos estiverem em um hiperplano de k dimensões para* $k < n$ *pontos.*

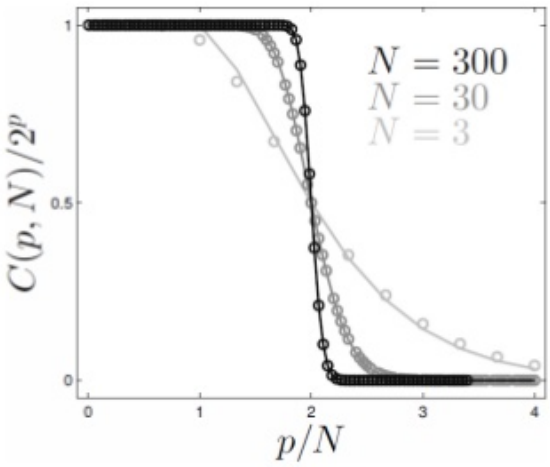

Figura 2: Fração de conjuntos de treinamento linearmente separáveis como função do número de exemplos por dimensão do espaço.

Um conjunto de m vetores $\{\boldsymbol{x}_\mu\}$ amostrados de forma independente de uma distribuição geradora de dados $p(\boldsymbol{x})$ estarão em posição geral quase sempre (i.e., a menos de um conjunto de medida nula). Em um relatório técnico do Systems Theory Laboratory de Stanford publicado em 1964 com o título *Geometrical and Statistical Properties of Linear Threshold Devices*, Thomas M. Cover demonstrou o seguinte teorema.

Teorema 1 (Teorema de Contagem de Funções de Cover). *Consideremos m pontos $\{\boldsymbol{x}_\mu\}$ com $\boldsymbol{x}_\mu \in \mathbb{R}^n$ e $\mu \in [m]$. Suponha que estes pontos estejam em posição geral. Seja $C(m, n)$ o número de conjuntos de treinamento linearmente separáveis construídos a partir destes m pontos em n dimensões, então*

$$C(m, n) = 2 \sum_{i=0}^{n-1} \binom{m-1}{i}$$

Demonstração. Primeiro construímos uma recursão utilizando um argumento simples. Começamos com m pontos. O número de conjuntos linearmente separáveis que podem ser construídos com esses pontos é $C(m, n)$. Agora adicionamos um novo ponto. Tomemos um particular hiperplano de separação.

Podem ocorrer um de dois cenários: o novo ponto está contido no hiperplano ou o ponto está em um dos semi-espaços. No primeiro caso, pequenos deslocamentos podem colocar o novo ponto de um lado ou do outro do plano e este ponto dá origem a dois conjuntos de treinamento distintos. Como um ponto está contido no hiperplano, o número de conjuntos linearmente separáveis nesse caso é $C(m, n-1)$, já que neste cenário temos um grau de liberdade a menos. No segundo caso, a adição do novo ponto define apenas um conjunto de treinamento. Escrevemos assim a seguinte relação de recorrência:

$$C(m+1, n) = C(m, n) + C(m, n-1)$$

Iterando a recursão demonstra-se o teorema. □

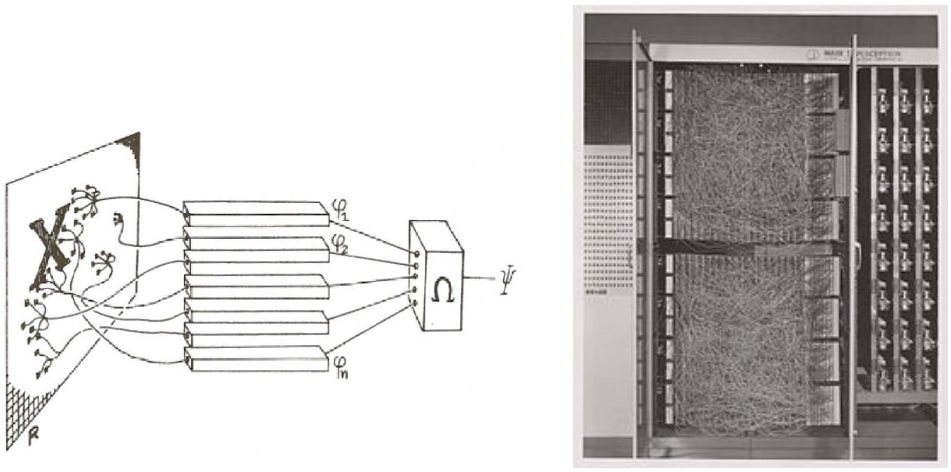

Figura 3: O Perceptron Mark I (à esquerda) foi a primeira implementação do algoritmo Perceptron produzida por Frank Rosenblatt em 1957 no Cornell Aeronautical Laboratory. O sistema era conectado à uma matriz com 400 fotocélulas de sulfeto de cádmio (à esquerda) que liam imagens apresentadas por uma matriz semelhante iluminada. No canto direito da foto à direita temos uma um conjunto de potenciômetros que implementavam os pesos adaptativos $\boldsymbol{W}$. O algoritmo de aprendizagem era implementado por motores elétricos que giravam os potenciômetros a cada exemplo apresentado.

O número de conjuntos de treinamento possíveis, linearmente separáveis ou não, é 2^m. Uma quantidade interessante é a fração de conjuntos de tamanho m linearmente separáveis, ou seja, $C(m, n)/2^m$.

Note que $C(2n, n) = 2^{m-1}$ e que se $m \leq n$, $C(n, m) = 2^m$. Para n grande podemos escrever

$$\frac{C(m, n)}{2^m} \approx \frac{1}{2}\left\{1 + \mathrm{Erf}\left[\sqrt{\frac{m}{2}}\left(\frac{2n}{p} - 1\right)\right]\right\}. \tag{1}$$

Na figura 2 mostramos o comportamento de fração de conjuntos de tamanho m linearmente separáveis como função de m/n para vários n. A fração colapsa abruptamente em $m = 2n$.

Considerando o regime em que n é grande, conjuntos de treinamento menores que $2n$ são linearmente separáveis quase sempre. Já conjuntos de treinamento com mais de $2n$ exemplos quase nunca são linearmente separáveis. Dessa forma o teorema sugere que um conjunto pode ser feito linearmente separável quando embebido em um espaço tal que $n > p/2$. O teorema também sugere que $2n$ exemplos são suficientes para identificar a regra de classificação, daí referir-se a este valor especial como a *capacidade* do Perceptron.

4 Algoritmo Perceptron de Rosenblatt

Em 1957 Frank Rosenblatt publicou um relatório técnico no Cornell Aeronautical Laboratory intitulado *The Perceptron: a perceiving and recognizing automaton*. O projeto era financiado pela US Navy e virou notícia no New York Times em 1958. Neste artigo Rosenblatt apresentou o primeiro algoritmo de aprendizagem de máquina e o implementou em FORTRAN num IBM 704. Em seguida o Perceptron seria implementado em hardware como o Mark I Perceptron.

Algorithm 1: Perceptron de Rosenblatt

$\boldsymbol{W} \leftarrow 0$
while $\exists l$: $(\boldsymbol{x}_l, y_l) \in S$ *e* $y_l \boldsymbol{W} \cdot \boldsymbol{x}_l \leq 0$ do
  $\boldsymbol{W} \leftarrow \boldsymbol{W} + y_l \boldsymbol{x}_l$
end
return $\boldsymbol{W}$

Assim o algoritmo Perceptron de Rosenblatt consiste em alterar o vetor de parâmetros apenas para casos classificados incorretamente de forma a corrigir esta classificação. A cada atualização dos parâmetros é possível que casos previamente classificados corretamente sejam modificados na direção errada. Em seu trabalho de 58, Rosenblatt também demonstrou a convergência do algoritmo para regras linearmente separáveis.

Teorema 2 (Convergência do algoritmo Perceptron). *Suponhamos que:*

1. *existe um* $\boldsymbol{T} \in \mathbb{S}^n(1)$ *tal que* $y_\mu(\boldsymbol{T} \cdot \boldsymbol{x}_\mu) > \gamma$ *para algum* $\gamma > 0$ *e para todos* $(\boldsymbol{x}_\mu, y_\mu)$ *no conjunto de treinamento* S *de tamanho* m.
2. $\|\boldsymbol{x}_\mu\| \leq R$.

Nessas condições o algoritmo Perceptron de Rosenblatt encontra $\boldsymbol{T}$ *em no máximo* $\lfloor R^2/\gamma^2 \rfloor$ *atualizações.*

Demonstração. Considerando que no passo $k \leq 1$ o exemplo $\boldsymbol{x}_l$ esteja classificado incorretamente temos que

$$\begin{aligned}
\|\boldsymbol{W}_{k+1}\| &= \|\boldsymbol{W}_{k+1}\|\|\boldsymbol{T}\| \ \text{(da suposição 1)} \\
&\geq \boldsymbol{W}_{k+1} \cdot \boldsymbol{T} \\
&= (\boldsymbol{W}_k + y_l \boldsymbol{x}_l) \cdot \boldsymbol{T} \\
&= \boldsymbol{W}_k \cdot \boldsymbol{T} + y_l \boldsymbol{x}_l \cdot \boldsymbol{T} \\
&> \boldsymbol{W}_k \cdot \boldsymbol{T} + \gamma \ \text{(da suposição 1)} \\
&> k\gamma \ \text{(por indução da suposição 1)}
\end{aligned}$$

Por outro lado,

$$\begin{aligned}
\|\boldsymbol{W}_{k+1}\|^2 &= \|\boldsymbol{W}_k + y_l \boldsymbol{x}_l\|^2 \ \text{(usando a suposição 1)} \\
&= \|\boldsymbol{W}_k\|^2 + \|y_l \boldsymbol{x}_l\|^2 + 2y_l(\boldsymbol{W}_k \cdot \boldsymbol{x}_l) \\
&\leq \|\boldsymbol{W}_k\|^2 + \|\boldsymbol{x}_l\|^2 \ \text{(classificação de } l \text{ está incorreta e } y_l \in \{\pm 1\}) \\
&\leq \|\boldsymbol{W}_k\|^2 + R^2 \ \text{(da suposição 2)} \\
&\leq kR^2 \ \text{(por indução da suposição 2)}
\end{aligned}$$

Tesmos então que $k^2\gamma < \|\boldsymbol{W}_{k+1}\|^2 \leq kR^2$, o que implica em $k < R^2/\gamma^2$. □

4.1 Algoritmo Perceptron como um sistema dinâmico

Um exercício prolífico consiste em olharmos para o limite (termodinâmico) $n \to \infty$ representando o algoritmo Perceptron como um sistema dinâmico em tempo contínuo. Começamos por adotar o Perceptron Booleano na forma $y(\boldsymbol{x}) = \Theta(\boldsymbol{T}\cdot\boldsymbol{x})$. Em seguida escrevemos um passo de atualização proporcionado pelo par $(\boldsymbol{x}_l, y_l)$:

$$\begin{aligned}\Delta \boldsymbol{W} &= \boldsymbol{W}(t+\varepsilon) - \boldsymbol{W}(t) \\ &= \varepsilon \boldsymbol{x}_l \left[y_l - \Theta(\boldsymbol{W}(t)\cdot \boldsymbol{x}_l) \right]\end{aligned}$$

Em l passos teremos:

$$\boldsymbol{W}(t + l\varepsilon + \varepsilon) - \boldsymbol{W}(t + l\varepsilon) = \varepsilon \boldsymbol{x}_l \left[y_l - \Theta(\boldsymbol{W}(t)\cdot \boldsymbol{x}_l + \mathscr{O}(l\varepsilon)) \right]$$

Aqui $\mathscr{O}(l\varepsilon)$ representa o resultado das atualizações até o passo l dado por $\boldsymbol{W}(t + l\varepsilon) = \boldsymbol{W}(t) + \mathscr{O}(l\varepsilon)$. Somando sobre l obtemos:

$$\boldsymbol{W}(t + n\varepsilon) - \boldsymbol{W}(t) = \sum_{l=0}^{n-1} \left[\boldsymbol{W}(t + l\varepsilon + \varepsilon) - \boldsymbol{W}(t + l\varepsilon) \right]$$

Tomando os limites simultâneos $n \to \infty$ e $n\varepsilon \to 0$ obtemos:

$$\lim_{n\to\infty, n\varepsilon\to 0} \frac{\boldsymbol{W}(t + n\varepsilon) - \boldsymbol{W}(t)}{n\varepsilon} = \lim_{n\to\infty, n\varepsilon\to 0} \frac{1}{n} \sum_{l=0}^{n-1} \boldsymbol{x}_l \left[y_l - \Theta(\boldsymbol{W}(t)\cdot \boldsymbol{x}_l + \mathscr{O}(l\varepsilon)) \right]$$

Finalmente obtemos um sistema dinâmico em tempo contínuo representando o algoritmo Perceptron:

$$\frac{d\boldsymbol{W}}{dt}(t) = \left\langle \boldsymbol{x} \left[y(\boldsymbol{x}) - \Theta(\boldsymbol{W}(t)\cdot \boldsymbol{x}) \right] \right\rangle_{p(\boldsymbol{x})}$$

Teorema 3 (Convergência do algoritmo Perceptron em tempo contínuo). *Se* $y(\boldsymbol{x}) = \Theta(\boldsymbol{T}\cdot\boldsymbol{x})$ *com* $\|\boldsymbol{T}\| = 1$, $\boldsymbol{x} \in X$ *e* $|X| < \infty$, *então*

$$\frac{d\boldsymbol{W}}{dt} = \left\langle \boldsymbol{x} \left[y(\boldsymbol{x}) - \Theta(\boldsymbol{W}\cdot \boldsymbol{x}) \right] \right\rangle_{p(\boldsymbol{x})}$$

converge, para t suficientemente grande, o ponto fixo $\boldsymbol{W}^*$ *com* $E_G[\boldsymbol{W}^*] = 0$.

Demonstração. A dinâmica pode ser reescrita como a descida pelo gradiente

$$\frac{d\boldsymbol{W}}{dt} = -\nabla E(\boldsymbol{W})$$

na superfície de erro

$$E(\boldsymbol{W}) := \left\langle (\boldsymbol{W}\cdot \boldsymbol{x}) \left[y(\boldsymbol{x}) - \Theta(\boldsymbol{W}\cdot \boldsymbol{x}) \right] \right\rangle_{p(\boldsymbol{x})}$$

Podemos demonstrar que $E(\boldsymbol{W})$ é uma função de Lyapunov. Temos que $E(\mathbf{0}) = 0$. Também que

$$\begin{aligned} E(\boldsymbol{W}) &= \langle (\boldsymbol{W}\cdot\boldsymbol{x})\left[\Theta(\boldsymbol{T}\cdot\boldsymbol{x}) - \Theta(\boldsymbol{W}\cdot\boldsymbol{x})\right]\rangle_{p(\boldsymbol{x})} \\ &= \left\langle |\boldsymbol{W}\cdot\boldsymbol{x}|\left[1 - \text{sign}(\boldsymbol{T}\cdot\boldsymbol{x})\text{sign}(\boldsymbol{W}\cdot\boldsymbol{x})\right]\right\rangle_{p(\boldsymbol{x})} \geq 0 \end{aligned}$$

Finalmente

$$\frac{dE}{dt} = \nabla E \cdot \frac{d\boldsymbol{W}}{dt} = -\|\nabla E\|^2 \leq 0$$

Especificamente para $|X| < \infty$:

$$\begin{aligned} \frac{dE}{dt} = -\left(\frac{d\boldsymbol{W}}{dt}\right)^2 &= -\langle (\tilde{\boldsymbol{x}}\cdot\boldsymbol{x})\left[\Theta(\boldsymbol{T}\cdot\tilde{\boldsymbol{x}}) - \Theta(\boldsymbol{W}\cdot\tilde{\boldsymbol{x}})\right]\left[\Theta(\boldsymbol{T}\cdot\boldsymbol{x}) - \Theta(\boldsymbol{W}\cdot\boldsymbol{x})\right]\rangle_{p(\boldsymbol{x})p(\tilde{\boldsymbol{x}})} \\ &= -\sum_j \left\langle x_j\left[\Theta(\boldsymbol{T}\cdot\boldsymbol{x}) - \Theta(\boldsymbol{W}\cdot\boldsymbol{x})\right]\right\rangle^2_{p(\boldsymbol{x})} \\ &\leq -\left\langle \sum_j (T_j x_j)\left[\Theta(\boldsymbol{T}\cdot\boldsymbol{x}) - \Theta(\boldsymbol{W}\cdot\boldsymbol{x})\right]\right\rangle^2_{p(\boldsymbol{x})} \quad \text{(pela desigualdade de Schwartz e } \|\boldsymbol{T}\| = 1) \\ &= -\langle |\boldsymbol{T}\cdot\boldsymbol{x}|\left[\Theta(-(\boldsymbol{T}\cdot\boldsymbol{x})(\boldsymbol{W}\cdot\boldsymbol{x}))\right]\rangle^2_{p(\boldsymbol{x})} \\ &\leq -\left[\min_{\boldsymbol{x}\in X^*} p(\boldsymbol{x})|\boldsymbol{T}\cdot\boldsymbol{x}|\right]^2 = -K^2 \ \text{ com } X^*(\boldsymbol{W}) = \{\boldsymbol{x}\in X : (\boldsymbol{T}\cdot\boldsymbol{x})(\boldsymbol{W}\cdot\boldsymbol{x}) < 0\} \end{aligned}$$

$X^*(\boldsymbol{W})$ não é vazio se $E(\boldsymbol{W}) > 0$. Como $\frac{dE}{dt}(\boldsymbol{W}) \leq -K^2$ e $E(\boldsymbol{W}) \geq 0$, $E = 0$ para $t \geq \frac{E(\boldsymbol{W}(0))}{K^2}$. Consequentemente, para t suficientemente grande, E é constante e um ponto fixo $\boldsymbol{W}^*$ é atingido, pois $\frac{d\boldsymbol{W}}{dt}(t) = 0$. Como $|\boldsymbol{T}\cdot\boldsymbol{x}| > 0$ quase sempre, $\text{sign}(\boldsymbol{T}\cdot\boldsymbol{x}) = \text{sign}(\boldsymbol{W}^*\cdot\boldsymbol{x})$ quase sempre.

□

5 Análise do Algoritmo de Gibbs

O Algoritmo de Rosenblatt foi a primeira proposta de algoritmo com uma demonstração rigorosa de convergência. Há outras alternativas para algoritmos de treinamento. O desempenho de diferentes algoritmos pode ser comparado calculado-se a *Curva de Aprendizagem* $E_G(m)$. Esta curva pode ser avaliada no pior caso (ou seja, para os piores conjuntos de treinamento possíveis) ou no caso típico. Na década de 1990 a comunidade de Física Estatística desenvolveu intensamente técnicas não-rigorosas capazes de determinar Curvas de Aprendizagem para diversos algoritmos em Perceptrons simples ou multicamada (veja, por exemplo, [3] e [4]). Nesta seção discutimos a determinação destas curvas para um algoritmo que não é prático, mas é passível de análise detalhada.

5.1 Geometria da aprendizagem em Perceptrons

Para que possamos explorar a geometria da aprendizagem em Perceptrons vamos restringir nosso cenário para Percetrons Booleanos $y = \text{sign}(\boldsymbol{W}\cdot\boldsymbol{x})$. Os conjuntos de treinamento serão gerados por um per-

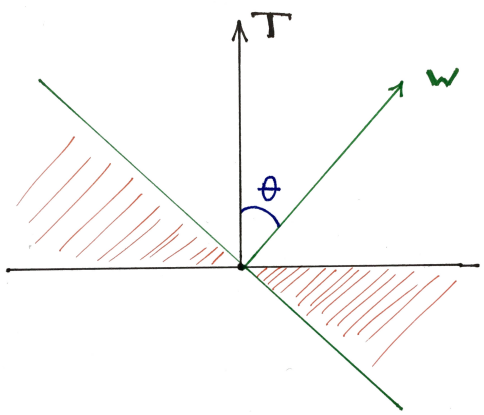

Figura 4: No caso realizável os dados são gerados por um Perceptron "professor" $\boldsymbol{T}$. O Perceptron que aprende a regra é $\boldsymbol{W}$ e entre ele e $\boldsymbol{T}$ há um ângulo $\boldsymbol{\theta}$. Na região hachurada há discordância na classificação.

ceptron $\boldsymbol{T}$ de forma que $\boldsymbol{W}, \boldsymbol{T} \in \mathbb{S}_{\sqrt{n}}$. Vamos assumir também que a distribuição geradora de casos é isotrópica sobre a hiperesfera $\mathbb{S}_{\sqrt{n}}$:

$$p(\boldsymbol{x}) = \prod_{j=1}^{n} \left[\frac{1}{2}\delta(x_j + 1) + \frac{1}{2}\delta(x_j - 1) \right]$$

O processo de aprendizagem neste cenário pode ser imaginado como um passeio do vetor $\boldsymbol{W}$ pela hiperesfera de raio $\sqrt{n}$. O erro de generalização pode ser calculado facilmente observando a figura 4. A região de discordância na classificação depende do menor dos ângulos entre os vetores. Se a distribuição de casos é uniforme pelos ângulos a probabilidade de discordância será:

$$\begin{aligned} E_G[\boldsymbol{W}] &= \mathbb{P}\left\{\operatorname{sign}(\boldsymbol{T}\cdot\boldsymbol{x})\operatorname{sign}(\boldsymbol{W}\cdot\boldsymbol{x}) \leq 0\right\} \\ &= \frac{\boldsymbol{\theta}}{\pi} \\ &= \frac{1}{\pi}\arccos R \end{aligned}$$

com o *overlap* definido como

$$R = \frac{\boldsymbol{T}\cdot\boldsymbol{W}}{n} = \cos(\pi E_G)$$

Definição 3 (Espaço de Versões). *O Espaço de Versões de um conjunto de treinamento* $S = \{(\boldsymbol{x}_\mu, y_\mu)\}_{\mu=1}^{m}$ *é o conjunto de regras compatíveis com S, ou seja*

$$\mathscr{V}(S) := \{\boldsymbol{x} : E_T[\boldsymbol{W}, S] = 0\}$$

.

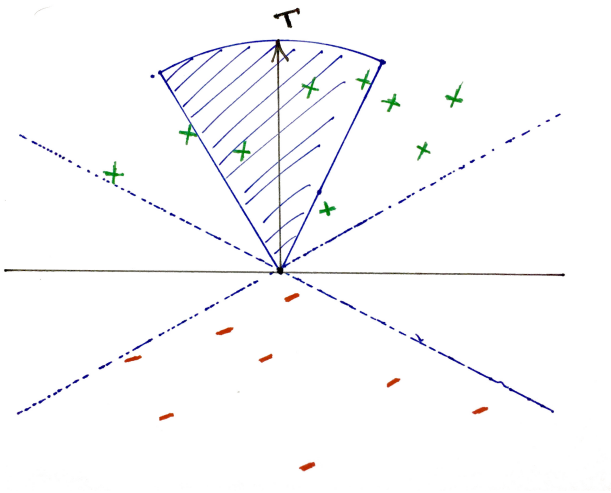

Figura 5: A região hachurada representa o Espaço de Versões, o conjunto de vetores compatíveis com o conjunto de treinamento.

5.2 Algoritmo de Gibbs

O algoritmo de Gibbs consiste em amostrarmos com distribuição uniforme uma regra no Espaço de versões, ou seja,

$$\boldsymbol{W} \sim \mathbb{U}[\mathscr{V}(S)].$$

O objetivo desta seção é calcular a curva de aprendizagem $E_G(\alpha)$ para $\alpha := \frac{m}{n}$ para $n \to \infty$. Neste limite, conhecido como *limite termodinâmico*, a curva de aprendizagem se torna independente da composição detalhada do conjunto de treinamento.

O erro de generalização é função do *overlap* R. Gostaríamos de determinar o *overlap* mais provável para um conjunto S de tamanho m. Como o algoritmo de Gibbs amostra uniformemente membros do espaço de versões, podemos em princípio determinar a probabilidade de um dado E_G avaliando a fração do espaço de versões ocupada por vetores que formam com $\boldsymbol{T}$ um ângulo πE_G. No limite termodinâmico a situação é consideravelmente mais simples já que é possível mostrar que quase todos (a menos de um conjunto com medida nula) vetores formam o mesmo ângulo com $\boldsymbol{T}$.

O cálculo da curva de aprendizagem típica do algoritmo de Gibbs que se segue não é rigoroso, mas pode ser verificado por simulação. De fato, como veremos, o cálculo depende de um passo heurístico no momento matematicamente injustificável.

5.3 Cálculo de Réplicas

O cálculo que introduzimos nessa seção foi proposta por Elizabeth Gardner em [1]. A ideia consiste em calcularmos a fração da hiperesfera $\mathbb{S}_{\sqrt{n}}$ ocupada pelo espaço de versões de um conjunto de treinamento S de tamanho m. Chamemos esta fração de $\Omega(S)$. Queremos determinar a fração típica. No limite ter-

modinâmico é possível mostrar que a distribuição $p(\Omega)$ induzida pelos conjuntos de treinamento está concentrada em torno de sua moda e que se calcularmos $\langle\Omega\rangle$ de fato obteremos um valor de Ω bastante raro. Para evitar este problema calculamos valor esperado de $\ln\Omega$. Além disso é possível demonstrar [2] que $\ln\Omega(S)$ é uma quantidade auto-mediante, ou seja, conforme $n \to \infty$ as flutuações induzidas pela aleatoriedade na escolha do conjunto de treinamento desaparecem.

No limite termodinâmico a esperança $\langle\ln\Omega(S)\rangle_{p(S)}$ assume a forma

$$\lim_{n\to\infty}\frac{1}{n}\langle\ln\Omega(S)\rangle_{p(S)} = \max_R \phi(R,\alpha).$$

ou seja, fixado o tamanho do conjunto de treinamento α, a fração da hiperesfera ocupada pelo espaço de versões tem a forma

$$\Omega(\alpha) \simeq e^{n\phi(\cos(\pi E_G(\alpha)),\alpha)},$$

onde $E_G(\alpha) = \frac{1}{\pi}\arccos(\operatorname{argmax}_R\phi(R,\alpha))$ é a curva de aprendizagem. A fração ocupada pelo espaço de versões é assim função apenas do tamanho do conjunto de treinamento e para cada α há um *overlap* $R(\alpha)$ dominante no espaço de versões. Essa é uma peculiaridade da hiperesfera com infinitas dimensões: o espaço de versões tem a forma de um "guarda-chuva" com todos os vetores formando o mesmo ângulo em relação à regra geradora dos dados $\boldsymbol{T}$.

Começamos por utilizar o seguinte limite:

$$\langle\ln\Omega(S)\rangle = \lim_{k\to 0}\frac{\langle\Omega^k(S)\rangle - 1}{k}.$$

A $\langle\Omega^k(S)\rangle$ chamamos forma replicada.

A fração da hiperesfera ocupada pelo espaço de versões é

$$\langle\Omega(S)\rangle = \left\langle \int \frac{d\boldsymbol{W}\delta(\|\boldsymbol{W}\|^2 - n)}{\int d\boldsymbol{B}\delta(\|\boldsymbol{B}\|^2 - n)} \prod_{\mu=1}^{m}\Theta\left(\frac{\boldsymbol{T}\cdot\boldsymbol{x_\mu}}{\sqrt{n}}\frac{\boldsymbol{W}\cdot\boldsymbol{x_\mu}}{\sqrt{n}}\right)\right\rangle_{\boldsymbol{T},\boldsymbol{x_\mu}}$$

Temos que

$$\int d\boldsymbol{B}\delta(\|\boldsymbol{B}\|^2 - n) = (2\pi e)^{n/2}$$

A forma replicada é

$$\begin{aligned}
\langle\Omega^k(S)\rangle &= \left\langle \int \frac{\prod_{a=1}^k d\boldsymbol{W}_a\delta(\|\boldsymbol{W}_a\|^2 - n)}{(2\pi e)^{nk/2}} \prod_{a=1}^{k}\prod_{\mu=1}^{m}\Theta\left(\frac{\boldsymbol{T}\cdot\boldsymbol{x_\mu}}{\sqrt{n}}\frac{\boldsymbol{W}_a\cdot\boldsymbol{x_\mu}}{\sqrt{n}}\right)\right\rangle_{\boldsymbol{T},\boldsymbol{x_\mu}} \\
&= \int \frac{\prod_{a=1}^k d\boldsymbol{W}_a\delta(\|\boldsymbol{W}_a\|^2 - n)}{(2\pi e)^{nk/2}} \\
&\times \int \prod_{a,\mu} d\lambda_{a\mu} \int \prod_{\mu} du_\mu \prod_{a,\mu}\Theta(\lambda_{a\mu}u_\mu)\left\langle \delta\left(\lambda_{a\mu} - \frac{\boldsymbol{W}_a\cdot\boldsymbol{x_\mu}}{\sqrt{n}}\right)\delta\left(u_\mu - \frac{\boldsymbol{T}\cdot\boldsymbol{x_\mu}}{\sqrt{n}}\right)\right\rangle
\end{aligned}$$

Neste passo utilizamos que $\int dx\, g(x)\delta(x-u) = g(u)$. Outra propriedade útil da distribuição de delta de Dirac é sua representação de Fourier

$$\delta(x) = \int \frac{d\hat{x}}{2\pi} e^{i\hat{x}x}.$$

Empregando a representação de Fourier escrevemos:

$$\begin{aligned}
\langle \Omega^k(S) \rangle &= \int \frac{\prod_{a=1}^{k} d\boldsymbol{W}_a \delta(\|\boldsymbol{W}_a\|^2 - n)}{(2\pi e)^{nk/2}} \\
&\times \int \prod_{a,\mu} \frac{d\lambda_{a\mu} d\hat{\lambda}_{a\mu}}{2\pi} \int \prod_{\mu} \frac{du_\mu d\hat{u}_\mu}{2\pi} \prod_{a,\mu} \Theta(\lambda_{a\mu} u_\mu) \exp\left[i \sum_{a,\mu} \lambda_{a\mu}\hat{\lambda}_{a\mu} + i \sum_{\mu} u_\mu \hat{u}_\mu \right] \\
&\times \underbrace{\left\langle \exp\left[-\frac{i}{\sqrt{n}} \sum_{a,\mu} \hat{\lambda}_{a\mu} \boldsymbol{W}_a \cdot \boldsymbol{x}_{\boldsymbol{\mu}} - \frac{i}{\sqrt{n}} \sum_{\mu} \hat{u}_\mu \boldsymbol{T} \cdot \boldsymbol{x}_{\boldsymbol{\mu}} \right] \right\rangle_{\boldsymbol{T}, \boldsymbol{x}_{\boldsymbol{\mu}}}}_{I}
\end{aligned}$$

Agora calculamos a esperança sobre os casos utilizando

$$p(\{\boldsymbol{x}_\mu\}) = \prod_{\mu=1}^{m} \prod_{j=1}^{n} \left[\frac{1}{2}\delta(x_{j\mu} + 1) + \frac{1}{2}\delta(x_{j\mu} - 1) \right]$$

$$\begin{aligned}
I &= \left\langle \prod_{\mu=1}^{m} \prod_{j=1}^{n} \sum_{x=\pm 1} \left[\frac{1}{2}\delta(x+1) + \frac{1}{2}\delta(x-1) \right] \exp\left[-\frac{i}{\sqrt{n}} \left(\sum_a \hat{\lambda}_{a\mu} W_{ja} + \hat{u}_\mu T_j \right) x \right] \right\rangle_{\boldsymbol{T}} \\
&= \left\langle \prod_{\mu,j} \cos\left[\frac{1}{\sqrt{n}} \left(\sum_a \hat{\lambda}_{a\mu} W_{ja} + \hat{u}_\mu T_j \right) \right] \right\rangle_{\boldsymbol{T}} \\
&= \left\langle \exp\left[\sum_{\mu,j} \ln \cos\left(\frac{1}{\sqrt{n}} \left(\sum_a \hat{\lambda}_{a\mu} W_{ja} + \hat{u}_\mu T_j \right) \right) \right] \right\rangle_{\boldsymbol{T}} \\
&\overset{n\to\infty}{\simeq} \left\langle \exp\left[-\frac{1}{2n} \sum_{\mu,j} \left(\sum_a \hat{\lambda}_{a\mu} W_{ja} + \hat{u}_\mu T_j \right)^2 \right] \right\rangle_{\boldsymbol{T}} \quad \text{(Série de Taylor para cos até ordem 2 e para ln em ordem 1)} \\
&= \left\langle \exp\left[-\frac{1}{2} \sum_{\mu} \sum_{a,b} \hat{\lambda}_{a\mu}\hat{\lambda}_{b\mu} \frac{\boldsymbol{W}_a \cdot \boldsymbol{W}_b}{n} - \sum_{\mu} \sum_{a} \hat{\lambda}_{a\mu}\hat{u}_\mu \frac{\boldsymbol{W}_a \cdot \boldsymbol{T}}{n} - \frac{1}{2} \sum_{\mu} \hat{u}_\mu^2 \right] \right\rangle_{\boldsymbol{T}} \\
&= \int \prod_{a<b} n dq_{ab} \int \prod_a n dR_a \left\langle \prod_a \delta\left(\boldsymbol{W}_a \cdot \boldsymbol{T} - nR_a \right) \right\rangle_{\boldsymbol{T}} \prod_{a<b} \delta\left(\boldsymbol{W}_a \cdot \boldsymbol{W}_b - nq_{ab} \right) \\
&\times \exp\left[-\frac{1}{2} \sum_{\mu,a} (\hat{\lambda}_{a\mu})^2 - \frac{1}{2} \sum_{\mu,a\neq b} \hat{\lambda}_{a\mu}\hat{\lambda}_{b\mu} q_{ab} - \sum_{\mu,a} \hat{\lambda}_{a\mu}\hat{u}_\mu R_a - \frac{1}{2} \sum_{\mu} \hat{u}_\mu^2 \right]
\end{aligned}$$

Retornando ao cálculo original:

$$
\begin{aligned}
\langle \Omega^k(S) \rangle &= \int \frac{\prod_{a=1}^{k} d\boldsymbol{W}_a \delta(\|\boldsymbol{W}_a\|^2 - n)}{(2\pi e)^{nk/2}} \prod_{a<b} \delta\left(\boldsymbol{W}_a \cdot \boldsymbol{W}_b - n q_{ab}\right) \left\langle \prod_a \delta\left(\boldsymbol{W}_a \cdot \boldsymbol{T} - n R_a\right) \right\rangle_{\boldsymbol{T}} \\
&\times \int \prod_{a<b} n dq_{ab} \int \prod_a n dR_a \int \prod_{a,\mu} \frac{d\lambda_{a\mu} d\hat{\lambda}_{a\mu}}{2\pi} \int \prod_\mu \frac{du_\mu d\hat{u}_\mu}{2\pi} \prod_{a,\mu} \Theta(\lambda_{a\mu} u_\mu) \\
&\times \exp\left[i \sum_{a,\mu} \lambda_{a\mu} \hat{\lambda}_{a\mu} + i \sum_\mu u_\mu \hat{u}_\mu \right] \\
&\times \exp\left[-\frac{1}{2} \sum_{\mu,a} (\hat{\lambda}_{a\mu})^2 - \frac{1}{2} \sum_{\mu,a\neq b} \hat{\lambda}_{a\mu} \hat{\lambda}_{b\mu} q_{ab} - \sum_{\mu,a} \hat{\lambda}_{a\mu} \hat{u}_\mu R_a - \frac{1}{2} \sum_\mu \hat{u}_\mu^2 \right]
\end{aligned}
$$

Agora introduzimos a representação de Fourier para as três distribuições delta remanescentes e escrevemos:

$$
\begin{aligned}
\langle \Omega^k(S) \rangle &= \int \prod_{a,j} \frac{dW_{ja}}{\sqrt{2\pi e}} \int \prod_a \frac{d\hat{k}_a}{4\pi} \exp\left[-i \sum_a \frac{\hat{k}_a}{2} \sum_j (W_{ja})^2 + in \sum_a \frac{\hat{k}_a}{2} \right] \\
&\times \int \prod_{a<b} \frac{dq_{ab} d\hat{q}_{ab}}{2\pi/n} \exp\left[-i \sum_{a<b,j} \hat{q}_{ab} W_{ja} W_{jb} + in \sum_{a<b} \hat{q}_{ab} q_{ab} \right] \\
&\times \int \prod_a \frac{dR_a d\hat{R}_a}{2\pi/n} \left\langle \exp\left[-i \sum_{a,j} \hat{R}_a W_{ja} T_j + in \sum_a \hat{R}_a R_a \right] \right\rangle_{\boldsymbol{T}} \\
&\times \int \prod_{a,\mu} \frac{d\lambda_{a\mu} d\hat{\lambda}_{a\mu}}{2\pi} \int \prod_\mu \frac{du_\mu d\hat{u}_\mu}{2\pi} \prod_{a,\mu} \Theta(\lambda_{a\mu} u_\mu) \\
&\times \exp\left[i \sum_{a,\mu} \lambda_{a\mu} \hat{\lambda}_{a\mu} + i \sum_\mu u_\mu \hat{u}_\mu \right] \\
&\times \exp\left[-\frac{1}{2} \sum_{\mu,a} (\hat{\lambda}_{a\mu})^2 - \frac{1}{2} \sum_{\mu,a\neq b} \hat{\lambda}_{a\mu} \hat{\lambda}_{b\mu} q_{ab} - \sum_{\mu,a} \hat{\lambda}_{a\mu} \hat{u}_\mu R_a - \frac{1}{2} \sum_\mu \hat{u}_\mu^2 \right]
\end{aligned}
$$

Como o erro de generalização independe de $\boldsymbol{T}$, seu valor esperado é redundante e podemos escolher

$T_j = 1$. Rearranjando os termos obtemos:

$$\begin{aligned}
\langle \Omega^k(S) \rangle &= \int \prod_a \frac{d\hat{k}_a}{4\pi} \int \prod_{a<b} \frac{dq_{ab} d\hat{q}_{ab}}{2\pi/n} \int \prod_a \frac{dR_a d\hat{R}_a}{2\pi/n} \\
&\times \exp\left[i\frac{n}{2} \sum_a \hat{k}_a + in \sum_{a<b} \hat{q}_{ab} q_{ab} + in \sum_a \hat{R}_a R_a \right] \\
&\times \left\{ \int \prod_a \frac{dW_a}{\sqrt{2\pi e}} \exp\left[-i \sum_a \frac{\hat{k}_a}{2} (W_a)^2 - i \sum_{a<b} \hat{q}_{ab} W_a W_b - i \sum_a \hat{R}_a W_a \right] \right\}^n \\
&\times \int \prod_{a,\mu} \frac{d\lambda_{a\mu} d\hat{\lambda}_{a\mu}}{2\pi} \int \prod_\mu \frac{du_\mu}{\sqrt{2\pi}} \prod_{a,\mu} \Theta(\lambda_{a\mu} u_\mu) \\
&\times \exp\left[i \sum_{a,\mu} \lambda_{a\mu} \hat{\lambda}_{a\mu} - \frac{1}{2} \sum_{\mu,a} (\hat{\lambda}_{a\mu})^2 - \frac{1}{2} \sum_{\mu,a\neq b} \hat{\lambda}_{a\mu} \hat{\lambda}_{b\mu} q_{ab} \right] \\
&\times \prod_\mu \underbrace{\int \frac{d\hat{u}_\mu}{\sqrt{2\pi}} \exp\left[i u_\mu \hat{u}_\mu - \hat{u}_\mu \sum_a \hat{\lambda}_{a\mu} R_a - \frac{1}{2} \hat{u}_\mu^2 \right]}_{II}
\end{aligned}$$

A integral II sobre $\hat{u}_\mu$ pode ser calculada usando a identidade

$$\int_{\mathbb{R}} \frac{dx}{\sqrt{2\pi}} \exp\left[-ax^2 + bx \right] = \frac{1}{\sqrt{2a}} \exp\left[\frac{b^2}{4a} \right]$$

O resultado é

$$II = \exp\left[-\frac{1}{2} u_\mu^2 - i u_\mu \sum_a \hat{\lambda}_{a\mu} R_a + \frac{1}{2} \sum_a (\hat{\lambda}_{a\mu} R_a)^2 \right]$$

Notando que há $m = \alpha n$ cópias das integrais em μ variáveis, ficamos com

$$\begin{aligned}
\langle \Omega^k(S) \rangle &= \int \prod_a \frac{d\hat{k}_a}{4\pi} \int \prod_{a<b} \frac{dq_{ab} d\hat{q}_{ab}}{2\pi/n} \int \prod_a \frac{dR_a d\hat{R}_a}{2\pi/n} \\
&\times \exp\left[i\frac{n}{2} \sum_a \hat{k}_a + in \sum_{a<b} \hat{q}_{ab} q_{ab} + in \sum_a \hat{R}_a R_a \right] \\
&\times \left\{ \int \prod_a \frac{dW_a}{\sqrt{2\pi e}} \exp\left[-i \sum_a \frac{\hat{k}_a}{2} (W_a)^2 - i \sum_{a<b} \hat{q}_{ab} W_a W_b - i \sum_a \hat{R}_a W_a \right] \right\}^n \\
&\times \left\{ \int \frac{du e^{-\frac{1}{2}u^2}}{\sqrt{2\pi}} \int \prod_a \frac{d\lambda_a d\hat{\lambda}_a}{2\pi} \prod_a \Theta(\lambda_a u) e^{-iu \sum_a \hat{\lambda}_a R_a} \right. \\
&\times \left. \exp\left[i \sum_a \lambda_a \hat{\lambda}_a - \frac{1}{2} \sum_a \left(1 - R_a^2\right) \hat{\lambda}_a^2 - \frac{1}{2} \sum_{a\neq b} \hat{\lambda}_a \hat{\lambda}_b \left(q_{ab} - R_a R_b \right) \right] \right\}^{\alpha n}
\end{aligned}$$

Definindo

$$\begin{aligned} G_S(\hat{k}_a, \hat{q}_{ab}, \hat{R}_a) &:= \ln\left[\int \prod_a \frac{dW_a}{\sqrt{2\pi e}} e^{-i\sum_a \frac{\hat{k}_a}{2}(W_a)^2 - i\sum_{a<b}\hat{q}_{ab}W_aW_b - i\sum_a \hat{R}_aW_a}\right] \\ &= -\frac{k}{2} - \frac{1}{2}\ln(\det\mathbb{A}) - \frac{1}{2}\sum_{a,b}\hat{R}_a\mathbb{A}^{-1}_{ab}\hat{R}_b \\ &= -\frac{k}{2} - \frac{1}{2}\mathrm{Tr}\ln\mathbb{A} - \frac{1}{2}\sum_{a,b}\hat{R}_a\mathbb{A}^{-1}_{ab}\hat{R}_b, \end{aligned}$$

com

$$\mathbb{A}_{ab} := i\delta_{ab}\hat{k}_a + i\hat{q}_{ab}(1-\delta_{ab})$$

e utilizando a identidade

$$\int \frac{d\boldsymbol{x}}{(2\pi)^{k/2}} e^{-\frac{1}{2}\boldsymbol{x}\cdot\mathbb{A}\boldsymbol{x}+\boldsymbol{x}\cdot\boldsymbol{b}} = \frac{e^{\frac{1}{2}\boldsymbol{b}\cdot\mathbb{A}^{-1}\boldsymbol{b}}}{\sqrt{\det\mathbb{A}}}.$$

Também definimos

$$G_E(q_{ab}, R_a) := \ln\left[\int Du \int \prod_a \frac{d\lambda_a d\hat{\lambda}_a}{2\pi} \prod_a \Theta(\lambda_a u) e^{-iu\sum_a \hat{\lambda}_a R_a + i\sum_a \lambda_a\hat{\lambda}_a - \frac{1}{2}\sum_a(1-R_a^2)\hat{\lambda}_a^2 - \frac{1}{2}\sum_{a\neq b}\hat{\lambda}_a\hat{\lambda}_b(q_{ab}-R_aR_b)}\right],$$

com $Du = du\, e^{-u^2/2}/\sqrt{2\pi}$, reescrevemos

$$\begin{aligned} \langle\Omega^k(S)\rangle &= \int \prod_a \frac{d\hat{k}_a}{4\pi} \int \prod_{a<b} \frac{dq_{ab}d\hat{q}_{ab}}{2\pi/n} \int \prod_a \frac{dR_a d\hat{R}_a}{2\pi/n} \\ &\times \exp\left\{n\left[i\frac{1}{2}\sum_a \hat{k}_a + i\sum_{a<b}\hat{q}_{ab}q_{ab} + i\sum_a \hat{R}_aR_a + G_S(\hat{k}_a, \hat{q}_{ab}, \hat{R}_a) + \alpha G_E(q_{ab}, R_a)\right]\right\} \\ &= \int \prod_a \frac{d\hat{k}_a}{4\pi} \int \prod_{a<b} \frac{dq_{ab}d\hat{q}_{ab}}{2\pi/n} \int \prod_a \frac{dR_a d\hat{R}_a}{2\pi/n} \\ &\times \exp\left\{n\left[\frac{1}{2}\mathrm{Tr}\mathbb{A}\mathbb{B} + i\sum_a \hat{R}_aR_a - \frac{k}{2} - \frac{1}{2}\mathrm{Tr}\ln\mathbb{A} - \frac{1}{2}\sum_{a,b}\hat{R}_a\mathbb{A}^{-1}_{ab}\hat{R}_b + \alpha G_E(q_{ab}, R_a)\right]\right\} \end{aligned}$$

Definindo $\mathbb{B}_{ab} := \delta_{ab} + q_{ab}(1-\delta_{ab})$. No limite $n\to\infty$ esta integral é (veja, por exemplo, [5])

$$\langle\Omega^k(S)\rangle \sim \exp\left\{n \operatorname*{Extr}_{R,\hat{R},\mathbb{A},q}\left[\frac{1}{2}\mathrm{Tr}\mathbb{A}\mathbb{B} + i\sum_a \hat{R}_aR_a - \frac{k}{2} - \frac{1}{2}\mathrm{Tr}\ln\mathbb{A} - \frac{1}{2}\sum_{a,b}\hat{R}_a\mathbb{A}^{-1}_{ab}\hat{R}_b + \alpha G_E(q_{ab}, R_a)\right]\right\}$$

Calculando derivadas em relação a $\hat{R}_a$ e $\mathbb{A}_{ab}$ e igualando-as a zero, encontramos

$$\hat{R}_a = i\sum_b \mathbb{A}_{ab}R_b$$

e

$$\mathbb{M}_{ab} := \mathbb{A}_{ab}^{-1} = \mathbb{B}_{ab} - R_a R_b$$

Chegamos assim a

$$\langle \Omega^k(S) \rangle \sim \exp\left\{ n \underset{R,q}{\text{Extr}} \left[\frac{1}{2} \text{Tr} \ln \mathbb{M} + \alpha G_E(q_{ab}, R_a) \right] \right\}$$

Seguimos em frente introduzindo o *Ansatz de réplica simétrica* $q_{ab} := q$ e $R_a := R$ introduzido por G. Parisi (trabalho pelo qual ganhou a medalha Dirac de 2002 [6]) e justificado a posteriori pela concordância das curvas de aprendizagem que obteremos com os resultados numéricos. $\mathbb{M}$ é uma matrix $k \times k$ que no Ansatz simétrico é

$$\mathbb{M} = \begin{bmatrix} 1 - R^2 & q - R^2 & \dots & q - R^2 \\ q - R^2 & 1 - R^2 & \dots & q - R^2 \\ \vdots & \vdots & \ddots & \vdots \\ q - R^2 & q - R^2 & \dots & 1 - R^2 \end{bmatrix}$$

Esta matriz apresenta $k - 1$ autovalores $1 - q$ e o autovalor remanescente $1 - q + k(q - R^2)$. Usando este resultado encontramos

$$\langle \Omega^k(S) \rangle \sim \exp\left\{ n \underset{R,q}{\text{Extr}} \left[\frac{k}{2} \ln(1-q) + \frac{1}{2} \ln\left(1 + k \frac{q - R^2}{1 - q} \right) + \alpha G_E(q, R) \right] \right\}$$

Agora trabalhamos com $G_E(q, R)$. Começamos por considerar que $\Theta(x) = 1$ para $x > 0$, usamos as simetrias das integrais e completamos o somatório com a condição $a \neq b$ para escrevermos:

$$\begin{aligned} G_E(q,R) &= \ln\left[\int Du \int \prod_a \frac{d\lambda_a d\hat{\lambda}_a}{2\pi} \prod_a \Theta(\lambda_a u) e^{-iuR\sum_a \hat{\lambda}_a + i\sum_a \lambda_a \hat{\lambda}_a - \frac{1}{2}(1-R^2)\sum_a \hat{\lambda}_a^2 - \frac{1}{2}(q-R^2)\sum_{a\neq b} \hat{\lambda}_a \hat{\lambda}_b} \right] \\ &= \ln\left[2\int_0^\infty Du \int_0^\infty \prod_a d\lambda_a \int \prod_a \frac{d\hat{\lambda}_a}{2\pi} e^{-iuR\sum_a \hat{\lambda}_a + i\sum_a \lambda_a \hat{\lambda}_a - \frac{1}{2}(1-q)\sum_a \hat{\lambda}_a^2 - \frac{1}{2}(q-R^2)\left(\sum_a \hat{\lambda}_a\right)^2} \right] \end{aligned}$$

Em seguida usamos a transformação de Hubbard-Stratonovich para eliminarmos o termo $\left(\sum_a \hat{\lambda}_a\right)^2$ e integrarmos em $\hat{\lambda}$:

$$\begin{aligned} G_E(q,R) &= \ln\left[2\int Dt \int_0^\infty Du \int_0^\infty \prod_a d\lambda_a \int \prod_a \frac{d\hat{\lambda}_a}{2\pi} e^{-\frac{1}{2}(1-q)\sum_a \hat{\lambda}_a^2 - i\sum_a \hat{\lambda}_a \left(\lambda_a - uR - t\sqrt{q-R^2}\right)} \right] \\ &= \ln\left\{ 2\int Dt \int_0^\infty Du \left[\int_0^\infty \frac{d\lambda}{2\pi(1-q)} e^{-\frac{1}{2(1-q)}\left(\lambda - uR - t\sqrt{q-R^2}\right)^2} \right]^k \right\} \\ &= \ln\left\{ 2\int Dt \int_0^\infty Du\, H^k\left(-\frac{uR + t\sqrt{q-R^2}}{\sqrt{1-q}} \right) \right\}, \end{aligned}$$

onde

$$H(x) := \int_x^\infty Dx$$

Mudando variáveis com

$$z = \frac{t\sqrt{q - R^2} + uR}{\sqrt{q}}$$

e integrando sobre u obtemos

$$G_E(q, R) = \ln \left\{ 2 \int Dz\, H\left(\frac{-Rz}{\sqrt{q - R^2}} \right)\, H^k\left(-z\sqrt{\frac{q}{1-q}} \right) \right\}$$

Até aqui todos os passos do cálculo foram rigorosos ou podem ser feitos rigorosos. No próximo passo fazemos uso de uma heurística se nenhuma justificativa matemática, mas que permite que a solução correta seja encontrada: fazemos uma expansão de Taylor em torno de $k = 0$ (que até agora era um número natural!):

$$\begin{aligned} \langle \Omega^k(S) \rangle &\sim e^{n \underset{R,q}{\text{Extr}} \left[\frac{k}{2}\ln(1-q) + \frac{1}{2}\ln\left(1 + k\frac{q-R^2}{1-q}\right) + \alpha \ln\left(2\int Dz\, H\left(\frac{-Rz}{\sqrt{q-R^2}}\right) H^k\left(-z\sqrt{\frac{q}{1-q}}\right)\right)\right]} \\ &\underset{k\to\infty}{\sim} e^{nk \underset{R,q}{\text{Extr}} \left[\frac{1}{2}\ln(1-q) + \frac{1}{2}\frac{q-R^2}{1-q} + 2\alpha \int Dz\, H\left(\frac{-Rz}{\sqrt{q-R^2}}\right) \ln H\left(-z\sqrt{\frac{q}{1-q}}\right)\right]} \end{aligned}$$

Finalmente calculamos o limite $k \to 0$ e encontramos

$$\begin{aligned} \frac{1}{n}\langle \ln \Omega(S) \rangle &= \lim_{k\to 0} \frac{\langle \Omega^k(S) \rangle - 1}{k} \\ &\sim \underset{R,q}{\text{Extr}} \left[\frac{1}{2}\ln(1-q) + \frac{1}{2}\frac{q - R^2}{1-q} + 2\alpha \int Dz\, H\left(\frac{-Rz}{\sqrt{q-R^2}} \right) \ln H\left(-z\sqrt{\frac{q}{1-q}} \right) \right] \end{aligned}$$

Calculando as derivadas em relação a R e q e igualando-as a zero obtemos duas equações para o extremo:

$$\begin{aligned} \frac{q - R^2}{1-q} &= \frac{\alpha}{\pi} \int Dz\, H\left(\frac{-Rz}{\sqrt{q-R^2}} \right) \frac{\exp\left[-z^2 \frac{q}{1-q} \right]}{H^2\left(-z\sqrt{\frac{q}{1-q}} \right)} \\ \frac{R\sqrt{q-R^2}}{\sqrt{q}\sqrt{1-q}} &= \frac{\alpha}{\pi} \int Dz \frac{\exp\left[-\frac{z^2}{2}\left(\frac{R^2}{q-R^2} + \frac{q}{1-q} \right) \right]}{H\left(-z\sqrt{\frac{q}{1-q}} \right)} \end{aligned}$$

O algoritmo de Gibbs funciona escolhendo um vetor $\boldsymbol{W}$ com distribuição uniforme sobre o espaço de versões do conjunto de treinamento S. R representa o *overlap* típico entre o vetor escolhido pelo algoritmo e o professor $\boldsymbol{T}$. q representa o *overlap* típico entre dois vetores escolhidos independentemente pelo algoritmo. Como o cenário que estamos analisando é realizável, o professor também é um membro do mesmo espaço de versões e há uma simetria entre as duas situações: 1. dado um professor no espaço de versões, escolhemos um vetor no mesmo espaço; 2. escolhemos um vetor no espaço de versões e, em

seguida, escolhemos um segundo vetor de forma independente. Dessa simetria esperamos que $q = R$ seja solução para as equações quando $q = R$:

$$R = \frac{\alpha}{\pi}\sqrt{1-R}\int Dz\,\frac{\exp\left[-\frac{Rz^2}{2}\right]}{H\left(-z\sqrt{R}\right)}$$

Resolvendo numericamente esta equação obtemos $R(\alpha)$ e daí a curva de aprendizagem $E_G(\alpha) = \frac{1}{\pi}\arccos[R(\alpha)]$. Na figura 6 mostramos a curva de aprendizagem típica de um Perceptron Booleano aprendendo por um algoritmo de Gibbs.

6 Aprendizado Bayesiano

Qual seria o melhor algoritmo possível no cenário que analisamos? A melhor classificação possível é obtida por um algoritmo bayesiano, por uma *Bayes Point Machine* [7]

No aprendizado bayesiano o teorema de Bayes fornece uma distribuição posterior sobre o espaço das regras após a observação do conjunto de treinamento S:

$$p(\boldsymbol{W}|S) = \frac{p(S|\boldsymbol{W})p(\boldsymbol{W})}{\int d\boldsymbol{W}p(S|\boldsymbol{W})p(\boldsymbol{W})}.$$

No caso livre de ruído a verossimilhança é dada por

$$p(S|\boldsymbol{W}) = \prod_{\mu=1}^{m}\mathbb{1}_{[y_\mu=\text{sign}(\boldsymbol{W}\cdot\boldsymbol{x}_\mu)]}$$

O objetivo do algoritmo é prescrever uma classificação y para um novo exemplo $\boldsymbol{x}$. A prescrição bayesiana é $y_{bayes} = \text{sign}(p(y=+1|S,\boldsymbol{x}) - p(y=-1|S,\boldsymbol{x}))$. Calculando o posterior $p(y|S,\boldsymbol{x})$ obtemos:

$$\begin{aligned}
p(y|S,\boldsymbol{x}) &= \int d\boldsymbol{W}\,p(\boldsymbol{W}|S)p(y|\boldsymbol{W},\boldsymbol{x}) \\
&= \int d\boldsymbol{W}\,\frac{p(S|\boldsymbol{W})p(\boldsymbol{W})}{\int d\boldsymbol{B}p(S|\boldsymbol{B})p(\boldsymbol{B})}p(y|\boldsymbol{W},\boldsymbol{x}) \\
&= \frac{\int_{\mathscr{V}(S)} d\boldsymbol{W}\,p(\boldsymbol{W})p(y|\boldsymbol{W},\boldsymbol{x})}{\int_{\mathscr{V}(S)} d\boldsymbol{B}p(\boldsymbol{B})} \\
&= \frac{\int_{\mathscr{V}(S)} d\boldsymbol{W}\,p(\boldsymbol{W})\mathbb{1}_{[y=\text{sign}(\boldsymbol{W}\cdot\boldsymbol{x})]}}{\int_{\mathscr{V}(S)} d\boldsymbol{B}p(\boldsymbol{B})}
\end{aligned}$$

Ou seja, o classificador bayesiano é resultado de uma votação no espaço de versões sobre a classificação do novo caso $\boldsymbol{x}$. No limite termodinâmico $n \to \infty$ é possível mostrar [7] que:

$$y_{bayes}(\boldsymbol{x}) = \text{sign}\left[\boldsymbol{W}_{CM}\cdot\boldsymbol{x}\right],$$

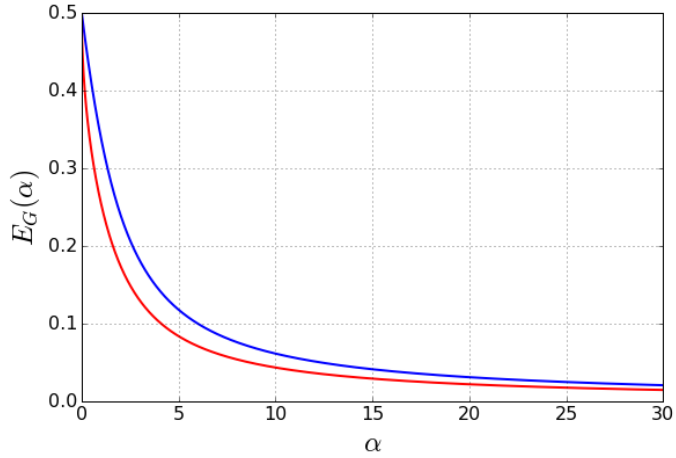

Figura 6: Curvas de aprendizagem típicas de um Perceptron Booleano em um cenário realizável com exemplos de treinamento gerados uniformemente em uma hiperesfera de $n \to \infty$ dimensões. A curva mais alta corresponde ao algoritmo de Gibbs e a mais baixa corresponde ao algoritmo ótimo bayesiano.

onde

$$\boldsymbol{W}_{CM} = \frac{\int_{\mathcal{V}(S)} d\boldsymbol{W}\, p(\boldsymbol{W})\boldsymbol{W}}{\int_{\mathcal{V}(S)} d\boldsymbol{B} p(\boldsymbol{B})}$$

é o centro de massa do espaço de versões. A máquina que implementa que encontra o centro de massa é a *Bayes Point Machine*. A curva de aprendizagem de uma *Bayes Point Machine* pode ser calculada usando técnicas semelhantes àquelas que apresentamos em detalhes na seção anterior (leia, por exemplo,[8]). O resultado pode ser escrito em termos do erro de generalização do algoritmo de Gibbs e é

$$E_G^{bayes}(alpha) = \frac{1}{\pi}\arccos\left[\sqrt{\cos(\pi E_G^{Gibbs}(\alpha))}\right]$$

Na figura 6 mostramos a curva de aprendizagem ótimo bayesiana juntamente com a curva de aprendizagem do algoritmo de Gibbs.

7 Problemas

1. Dado um conjunto de pontos $\{\boldsymbol{x}_j\}$ a casca convexa é o conjunto de todos os pontos $\boldsymbol{x}$ dados por $\boldsymbol{x} = \sum_j \alpha_j \boldsymbol{x}_j$, onde $\alpha_j \geq 0$ e $\sum_j \alpha_j = 1$. Considere um segundo conjunto de pontos $\{\boldsymbol{y}_j\}$ e sua casca convexa. Por definição, os dois conjuntos de pontos serão linearmente separáveis se existirem $\boldsymbol{W}$ e W_0 tais que $\boldsymbol{W} \cdot \boldsymbol{x_j} + W_0 > 0$ para todo $\boldsymbol{x}_j$ e $\boldsymbol{W} \cdot \boldsymbol{y_j} + W_0 < 0$ para todo $\boldsymbol{y_j}$. Mostre que se as cascas convexas se intersectarem, então os pontos não poderão ser linearmente separáveis. Mostre também que se os pontos forem linearmente separáveis, então suas cascas convexas não se intersectarão.

2. Mostre que para $n \to \infty$ e $m = \alpha n$

$$\frac{1}{n} \ln C(m, n) \begin{cases} = \alpha \ln 2 & \text{para } \alpha \leq 2 \\ \leq \alpha \ln \alpha - (\alpha - 1) \ln(\alpha - 1) & \text{para } \alpha > 2 \end{cases}$$

3. Gere numericamente m números aleatórios que sejam produtos de n números aleatórios independentes igualmente distribuídos entre 1 e 2 com m entre 10^3 e 10^6 e n entre 5 e 50. Aproxime a distribuição de x por um histograma e compare a evolução com n do valor mais provável x_{mp} de x, da média $\langle x \rangle$ e a quantidade $x_{typ} := e^{\langle \ln x \rangle}$. Construa um argumento para que x_{mp} coincida com x_{typ} quando $n \to \infty$. Por que $x_{mp} \leq x_{typ}$?

4. Mostre que no limite termodinâmico a maioria das regras no espaço de versões classifica um novo exemplo da mesma maneira que o centro de massa do espaço de versões. Para $m \to \infty$, verifique

$$\operatorname{sign}\left[\sum_{j=1}^{m} \operatorname{sign}(\boldsymbol{W}_j \cdot \boldsymbol{x})\right] = \operatorname{sign}\left[\sum_{j=1}^{m} \boldsymbol{W}_j \cdot \boldsymbol{x}\right]$$

Referências

[1] E. Gardner, The space of interactions in neural network models, *J. Phys A* (1988) 21:257.

[2] M. Talagrand, Self Averaging and teh Space of Interactions in Neural Networks *Random Struct. Alg.*(1999), 14:199

[3] T.L.H. Watkin, A. Rau e M. Biehl, The statistical mechanics of learning a rule *Reviews of Modern Physics* (1993), 65:499.

[4] A. Engel, C. Van den Broeck, *Statistical Mechanics of Learning*, Cambridge University Press, 2001.

[5] A. Erdélyi,*Asymptotic Expansions*, Dover Publications, 1956.

[6] G. Parisi,*The physical Meaning of Replica Symmetry Breaking*, arXiv:cond-mat/0205387 (2002)

[7] R. Herbrich, T. Graepel, C. Campbell, Bayes Point Machines, *Journal of Machine Learning Research* (2001) 1 245.

[8] M. Opper e D. Haussler, Generalization Performance of Bayes Optimal Classification Algorithm for Learning a Perceptron, *Physical Review Letters* (1991) 66 2677.