



DEPARTAMENTO DE  
**MATEMÁTICA**

## O poder da Aprendizagem Profunda

Felipe Kaminsky Riffel

Universidade Federal de Santa Catarina

23 de abril de 2025

## Artigo: Why does deep and cheap learning work so well?

Introdução

Expressabilidade e Eficiência de Redes Rasas

Custos de Achatamento

Interpretações físicas

## Referências

## Artigo: Why does deep and cheap learning work so well?

Introdução

Expressabilidade e Eficiência de Redes Rasas

Custos de Achatamento

Interpretações físicas

## Referências

Artigo:

LIN, Henry W.; TEGMARK, Max; ROLNICK, David. Why does deep and cheap learning work so well? Journal of Statistical Physics, v. 168, n. 6, p. 1223–1247, 2017.

## Artigo: Why does deep and cheap learning work so well?

Introdução

Expressabilidade e Eficiência de Redes Rasas

Custos de Achatamento

Interpretações físicas

## Referências

Três problemas principais da teoria de redes neurais:

- ▶ Expressabilidade: que funções podemos expressar?

Três problemas principais da teoria de redes neurais:

- ▶ Expressabilidade: que funções podemos expressar?
- ▶ Eficiência: quão complexa a rede tem que ser?

Três problemas principais da teoria de redes neurais:

- ▶ Expressabilidade: que funções podemos expressar?
- ▶ Eficiência: quão complexa a rede tem que ser?
- ▶ "Aprendibilidade": quão rápido a rede consegue aprender a ajustar os bons parâmetros? <sup>1</sup>

---

<sup>1</sup>Traduzido de "Learnability"



Três problemas principais da teoria de redes neurais:

- ▶ Expressabilidade: que funções podemos expressar?
- ▶ Eficiência: quão complexa a rede tem que ser?
- ▶ "Aprendibilidade": quão rápido a rede consegue aprender a ajustar os bons parâmetros? <sup>1</sup>

Aqui, focamos nos dois primeiros: **Expressabilidade** e **Eficiência**.

---

<sup>1</sup>Traduzido de "Learnability"

Problema: "como redes neurais funcionam bem na prática, se o número de funções possíveis é exponencialmente maior que o número de redes possíveis?"

**Exemplo:** imagem preta/branca de 1MP    vetor de 1000000  
entradas com 256 valores possíveis (valor em cada pixel)

**Exemplo:** imagem preta/branca de 1MP    vetor de 1000000  
entradas com 256 valores possíveis (valor em cada pixel)



$$\begin{pmatrix} x_1 \\ \vdots \\ x_{1000000} \end{pmatrix}$$

$$x_i \in I_{256} := \{1, 2, 3, \dots, 256\}$$

**Exemplo:** imagem preta/branca de 1MP    vetor de 1000000  
entradas com 256 valores possíveis (valor em cada pixel)



$$\begin{pmatrix} x_1 \\ \vdots \\ x_{1000000} \end{pmatrix}$$

$$x_i \in I_{256} := \{1, 2, 3, \dots, 256\}$$

Nº total de imagens possíveis:  $256^{1000000}$ .

**Exemplo:** imagem preta/branca de 1MP    vetor de 1000000  
entradas com 256 valores possíveis (valor em cada pixel)



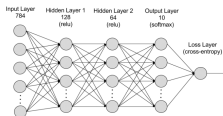
$$\begin{pmatrix} x_1 \\ \vdots \\ x_{1000000} \end{pmatrix}$$

$$x_i \in I_{256} := \{1, 2, 3, \dots, 256\}$$

Nº total de imagens possíveis:  $256^{1000000}$ .

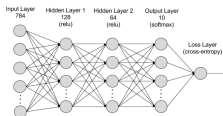
Se existe  $p : I_{256} \rightarrow (0, 1)$  que associa cada imagem a uma probabilidade,  $p$  deve ter uma lista  $256^{1000000}$  valores (!!!)

Porém, redes neurais relativamente simples conseguem calcular bem a tarefa.



$$p(\text{Gato}|\mathbf{x}) = 83\%$$

Porém, redes neurais relativamente simples conseguem calcular bem a tarefa.

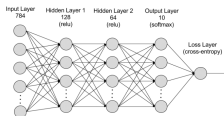


$$p(\text{Gato}|\mathbf{x}) = 83\%$$

A **matemática** ajuda a explicar: as redes neurais conseguem diminuir drasticamente a explosão combinatória de número de parâmetros em relação ao número de valores;



Porém, redes neurais relativamente simples conseguem calcular bem a tarefa.



$$p(\text{Gato}|\mathbf{x}) = 83\%$$

A **matemática** ajuda a explicar: as redes neurais conseguem diminuir drasticamente a explosão combinatória de número de parâmetros em relação ao número de valores;

A razão também é **física**: as leis sugerem que os datasets de interesse são, em sua maioria, advindos de distribuições simples.

## Artigo: Why does deep and cheap learning work so well?

Introdução

Expressabilidade e Eficiência de Redes Rasas

Custos de Achatamento

Interpretações físicas

## Referências

Considere  $\mathbf{x} \in \mathbb{R}^d$ . Sejam  $A_i : \mathbb{R}^d \rightarrow \mathbb{R}^d$  operadores afim, i.e.,

$$A_i = W_i - b_i$$

com  $W_i \in \mathbb{R}^{m_i \times n_i}$  e  $b_i \in \mathbb{R}^{n_i}$ .

Considere  $\mathbf{x} \in \mathbb{R}^d$ . Sejam  $A_i : \mathbb{R}^d \rightarrow \mathbb{R}^d$  operadores afim, i.e.,

$$A_i = W_i - b_i$$

com  $W_i \in \mathbb{R}^{m_i \times n_i}$  e  $b_i \in \mathbb{R}^{n_i}$ .

Dadas  $\sigma_i : \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{m_i}$  não linear, chamamos de rede neural feedforward uma função  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$  da forma:

$$\mathbf{f}(\mathbf{x}) = \sigma_k A_k \dots \sigma_2 A_2 \sigma_1 A_1 \mathbf{x}. \quad (1)$$

Considere  $\mathbf{x} \in \mathbb{R}^d$ . Sejam  $A_i : \mathbb{R}^d \rightarrow \mathbb{R}^d$  operadores afim, i.e.,

$$A_i = W_i - b_i$$

com  $W_i \in \mathbb{R}^{m_i \times n_i}$  e  $b_i \in \mathbb{R}^{n_i}$ .

Dadas  $\sigma_i : \mathbb{R}^{m_i} \rightarrow \mathbb{R}^{m_i}$  não linear, chamamos de rede neural feedforward uma função  $\mathbf{f} : \mathbb{R}^n \rightarrow \mathbb{R}^m$  da forma:

$$\mathbf{f}(\mathbf{x}) = \sigma_k A_k \dots \sigma_2 A_2 \sigma_1 A_1 \mathbf{x}. \quad (1)$$

- ▶ Aqui, se admite também  $\sigma_k = I$ ;
- ▶ Cada composição  $\sigma_i A_i$  é chamada de *camada* da rede;
- ▶ Cada componente da operação  $\sigma_{ij} A_{i,j} x$  (linha da matriz + aplicação de  $\sigma_i$ ) é chamado de *neurônio*;

$\sigma_i$  pode ser qualquer operador não linear. Escolhas comuns são, dado  $\mathbf{x} = (x_1, \dots, x_n)$ :

- ▶ Função local: escolha  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  não linear e aplique ponto a ponto  $\sigma_i(\mathbf{x}) = (\sigma(x_1), \dots, \sigma(x_n))$ ;

$\sigma_i$  pode ser qualquer operador não linear. Escolhas comuns são, dado  $\mathbf{x} = (x_1, \dots, x_n)$ :

- ▶ Função local: escolha  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  não linear e aplique ponto a ponto  $\sigma_i(\mathbf{x}) = (\sigma(x_1), \dots, \sigma(x_n))$ ;
- ▶ Max-pooling:  $\sigma_i(\mathbf{x}) = \max_{j=1, \dots, n}(x_j)$ ;

$\sigma_i$  pode ser qualquer operador não linear. Escolhas comuns são, dado  $\mathbf{x} = (x_1, \dots, x_n)$ :

- ▶ Função local: escolha  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  não linear e aplique ponto a ponto  $\sigma_i(\mathbf{x}) = (\sigma(x_1), \dots, \sigma(x_n))$ ;
- ▶ Max-pooling:  $\sigma_i(\mathbf{x}) = \max_{j=1, \dots, n}(x_j)$ ;
- ▶ Softmax:

$$\sigma_i(\mathbf{x}) = \frac{1}{\sum_{j=1}^n e^{x_j}} (e^{x_1}, \dots, e^{x_n}).$$



Seja  $\mathbf{f}$  rede neural da forma  $\mathbf{f}(\mathbf{x}) = A_2 \sigma A_1 \mathbf{x}$ , onde  $\sigma$  é aplicação não linear ponto a ponto qualquer. Considere as camadas de entrada, escondida e de saída com tamanhos 2, 4 e 1 respectivamente. Então,  $\mathbf{f}$  pode aproximar uma porta de multiplicação arbitrariamente bem.

Ou seja, dado  $\varepsilon > 0$ , para qualquer  $\sigma$  não linear (aplicada ponto a ponto), existem  $A_1 : \mathbb{R}^2 \rightarrow \mathbb{R}^4$ ,  $A_2 : \mathbb{R}^4 \rightarrow \mathbb{R}$  tais que a rede  $f(x) = A_2 \sigma A_1 \mathbf{x}$  é tal que, dado  $x = (u \ v)^T$  qualquer

$$|f(x) - uv| < \varepsilon$$

para  $u, v$  em um compacto qualquer.

Seja  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  não linear qualquer suficientemente suave. Na expansão de Taylor em torno de  $x = 0$ :

$$\sigma(u) = \sigma(0) + \sigma'(0)u + \frac{u^2}{2}\sigma''(0) + \mathcal{O}(u^3).$$

Sem perda de generalidade, considere  $\sigma''(0) \neq 0$  (ou então, ajuste  $b_1$  para que  $\sigma''(A_{1,1}x - b_{1,1}), \sigma''(A_{1,2}x - b_{1,2}) \neq 0$ , que deve existir dado que é não linear).

Seja  $\sigma : \mathbb{R} \rightarrow \mathbb{R}$  não linear qualquer suficientemente suave. Na expansão de Taylor em torno de  $x = 0$ :

$$\sigma(u) = \sigma(0) + \sigma'(0)u + \frac{u^2}{2}\sigma''(0) + \mathcal{O}(u^3).$$

Sem perda de generalidade, considere  $\sigma''(0) \neq 0$  (ou então, ajuste  $b_1$  para que  $\sigma''(A_{1,1}x - b_{1,1}), \sigma''(A_{1,2}x - b_{1,2}) \neq 0$ , que deve existir dado que é não linear).

Então,

$$\begin{aligned} m(u, v) &:= \frac{\sigma(u+v) + \sigma(-u-v) - \sigma(u-v) - \sigma(v-u)}{4\sigma''(0)} \\ &= \sigma''(0) \frac{(u+v)^2 + (-u-v)^2 - (u-v)^2 - (v-u)^2 + \mathcal{O}((u+v)^3)}{4\sigma''(0)} \\ &= uv + \mathcal{O}((u+v)^3) \end{aligned}$$

Ou seja,  $m(u, v) = uv + \mathcal{O}((u + v)^3)$ , de modo que

$$\lim_{u^2+v^2 \rightarrow 0} \frac{m(u, v) - uv}{u^2 + v^2} = 0.$$

Ou seja,  $m(u, v) = uv + \mathcal{O}((u + v)^3)$ , de modo que  
 $\lim_{u^2+v^2 \rightarrow 0} \frac{m(u,v)-uv}{u^2+v^2} = 0$ .

Veja que,  $m(u, v) = A_2 \sigma A_1 (u \ v)^T$ , onde:

$$A_1 = W_1 - b_1 = \begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} - b_1,$$

$$A_2 = W_2 = (4\sigma''(0))^{-1} \begin{pmatrix} 1 & 1 & -1 & -1 \end{pmatrix} - b_2.$$

Ou seja,  $m(u, v) = uv + \mathcal{O}((u + v)^3)$ , de modo que  
 $\lim_{u^2+v^2 \rightarrow 0} \frac{m(u,v)-uv}{u^2+v^2} = 0$ .

Veja que,  $m(u, v) = A_2 \sigma A_1 (u \ v)^T$ , onde:

$$A_1 = W_1 - b_1 = \begin{pmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \\ -1 & 1 \end{pmatrix} - b_1,$$

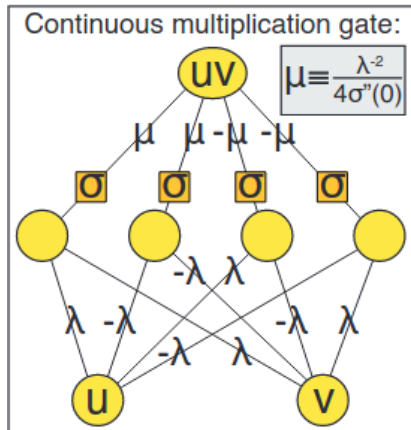
$$A_2 = W_2 = (4\sigma''(0))^{-1} \begin{pmatrix} 1 & 1 & -1 & -1 \end{pmatrix} - b_2.$$

Taylor fornece uma estimativa local, sendo boa para  $u, v \approx 0$ .  
Para  $u, v$  num compacto de raio qualquer, tome  $A_1 = \lambda W_1 - b_1$   
e  $A_2 = \lambda^{-2}W_2 - b_2$  na definição de  $\mathbf{f}$ , de modo a obter

$$f(x) = (\lambda^{-2}A_2)\sigma(\lambda A_1)x = \lambda^{-2}(\lambda u \lambda v) = uv,$$

tornando a estimativa tão boa quanto se queira.  $\square$

Figura: Ilustração da arquitetura da rede no teorema anterior



Fonte: Lin, et.al. (2017)



**Corolário:** Para cada polinômio multivariado e qualquer tolerância  $\varepsilon > 0$  existe uma rede neural de tamanho finito  $N$  (independente de  $\varepsilon$ ) que aproxima o polinômio a uma precisão melhor que  $\varepsilon$ . Além disso,  $N$  é limitado pela complexidade do polinômio, escalando conforme o número de multiplicações requeridas vezes um fator ligeiramente maior que 4.

Ideia: montamos uma rede neural em "blocos", onde cada produto pode ser representado por uma rede neural descrita no teorema anterior.

Cada produto necessita de 4 neurônios de camada escondida (a saída de um neurônio corresponde à entrada do seguinte). Ainda, temos neurônios a mais para os termos remanescentes, fazendo a passagem de uma camada para a outra, sem alterar o valor. O número de passagens é

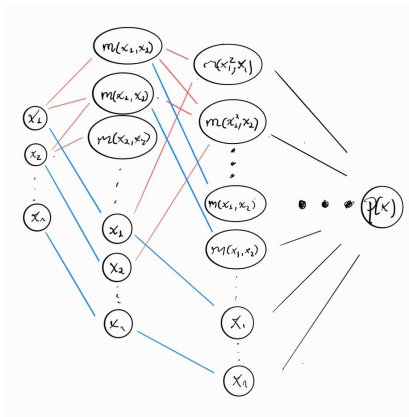
Ideia: montamos uma rede neural em "blocos", onde cada produto pode ser representado por uma rede neural descrita no teorema anterior.

Cada produto necessita de 4 neurônios de camada escondida (a saída de um neurônio corresponde à entrada do seguinte). Ainda, temos neurônios a mais para os termos remanescentes, fazendo a passagem de uma camada para a outra, sem alterar o valor. O número de passagens é

Para os termos remanescentes, podemos construir uma "camada de passagem"  $u \mapsto u$  com 1 neurônio, dado que:

$$u \approx \frac{\sigma(u) - \sigma(0)}{\sigma'(0)}$$

**Figura:** Ilustração da rede construída: em **vermelho**, conexões de produto (4 neurônios); em **azul**, camadas de passagem (1 neurônio)



Fonte: Autor.

Dadas  $n$  variáveis, conseguimos aproveitar cada camada anterior , pois para obter um polinômio:

- ▶ de grau 2, multiplicamos cada variável entre si e passamos as demais adiante, nos dando  $4n^2 + n \leq 5n^2$  neurônios;

Dadas  $n$  variáveis, conseguimos aproveitar cada camada anterior , pois para obter um polinômio:

- ▶ de grau 2, multiplicamos cada variável entre si e passamos as demais adiante, nos dando  $4n^2 + n \leq 5n^2$  neurônios;
- ▶ de grau 3, multiplicamos os  $n^2$  termos de grau 2 por cada variável e passamos os termos restantes, nos dando  $4n^3 +$ ;

Dadas  $n$  variáveis, conseguimos aproveitar cada camada anterior , pois para obter um polinômio:

- ▶ de grau 2, multiplicamos cada variável entre si e passamos as demais adiante, nos dando  $4n^2 + n \leq 5n^2$  neurônios;
- ▶ de grau 3, multiplicamos os  $n^2$  termos de grau 2 por cada variável e passamos os termos restantes, nos dando  $4n^3 +$ ;
- ▶  $\vdots$ ;
- ▶ de grau  $d$ , multiplicamos os  $n^{d-1}$  termos de grau  $d - 1$  entre as  $n$  variáveis e passamos os demais adiante, tendo  $4n^d + n^{d-1} + \dots + n \leq 5n^d$  neurônios;

Dadas  $n$  variáveis, conseguimos aproveitar cada camada anterior , pois para obter um polinômio:

- ▶ de grau 2, multiplicamos cada variável entre si e passamos as demais adiante, nos dando  $4n^2 + n \leq 5n^2$  neurônios;
- ▶ de grau 3, multiplicamos os  $n^2$  termos de grau 2 por cada variável e passamos os termos restantes, nos dando  $4n^3 +$ ;
- ▶  $\vdots$ ;
- ▶ de grau  $d$ , multiplicamos os  $n^{d-1}$  termos de grau  $d - 1$  entre as  $n$  variáveis e passamos os demais adiante, tendo  $4n^d + n^{d-1} + \dots + n \leq 5n^d$  neurônios;

No final, temos um número de neurônios da ordem de:

$$5n^2 + 5n^3 + \dots + 5n^d = \mathcal{O}(5n^d).$$



Se  $\mathbf{x} \in \{0, 1\}^n$ , temos  $x_i^2 = x_i$ , de modo que todo polinômio assume a forma

$$p(\mathbf{x}) = a_0 + \sum_i a_i x_i + \sum_{i < j} a_{ij} x_i x_j + \sum_{i < j < k} a_{ijk} x_i x_j x_k \cdots$$

No total, temos  $2^n$  termos distintos.

Ainda, qualquer produto de binários pode ser representado com um único neurônio na camada escondida com  $\sigma(x) = \frac{1}{1+e^{-x}}$ :

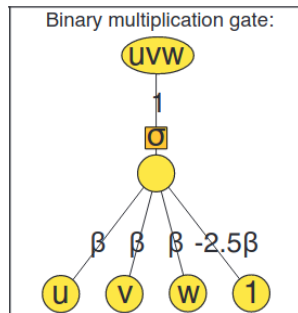
$$\prod_{i \in K} x_i = \lim_{\beta \rightarrow \infty} \sigma \left[ -\beta \left( k - \frac{1}{2} - \sum_{i \in K} x_i \right) \right],$$

pois  $\sigma(x) \rightarrow 0$  se  $x \rightarrow -\infty$  e  $\sigma(x) \rightarrow 1$  se  $x \rightarrow \infty$ .

Ainda, qualquer produto de binários pode ser representado com um único neurônio na camada escondida com  $\sigma(x) = \frac{1}{1+e^{-x}}$ :

$$\prod_{i \in K} x_i = \lim_{\beta \rightarrow \infty} \sigma \left[ -\beta \left( k - \frac{1}{2} - \sum_{i \in K} x_i \right) \right],$$

pois  $\sigma(x) \rightarrow 0$  se  $x \rightarrow -\infty$  e  $\sigma(x) \rightarrow 1$  se  $x \rightarrow \infty$ .



Assim, qualquer polinômio de  $n$  variáveis binárias pode ser representado por uma rede com:

- ▶ Uma camada de entrada, com  $n + 1$  neurônios;
- ▶ Uma camada escondida, com  $2^n$  neurônios (um para cada produto e termo livre);
- ▶ Uma camada de saída.

## Artigo: Why does deep and cheap learning work so well?

Introdução

Expressabilidade e Eficiência de Redes Rasas

Custos de Achatamento

Interpretações físicas

## Referências

**Exemplo:** Fatoração de matrizes

- ▶ FFT (Transformada rápida de Fourier);
- ▶ Matrizes de posto baixo:  $A = MN$ , com  $M_{n \times k}, N_{k \times n}$ ;
- ▶ Cholesky, LU, QR esparsas;

**Teorema:** Dados  $x_1, \dots, x_n \in \mathbb{R}$  e  $\sigma \in C^\infty$ , o monômio  $\prod_{i=1}^n x_i$  pode ser aproximado por uma rede neural de 1 camada com  $2^n$  neurônios, seguindo a fórmula

$$\prod_{i=1}^n x_i \approx \frac{1}{2^n} \sum_{\{s\}} s_1 \cdots s_n \sigma(s_1 x_1 + \cdots + s_n x_n),$$

onde  $s_i \in \{-1, 1\}$ , para cada  $i = 1, \dots, k$ , e a soma é tomada sobre todas as  $2^n$  configurações possíveis de  $s_1 \cdots s_n$ .

**Teorema:** Dados  $x_1, \dots, x_n \in \mathbb{R}$  e  $\sigma \in C^\infty$ , o monômio  $\prod_{i=1}^n x_i$  pode ser aproximado por uma rede neural de 1 camada com  $2^n$  neurônios, seguindo a fórmula

$$\prod_{i=1}^n x_i \approx \frac{1}{2^n} \sum_{\{s\}} s_1 \cdots s_n \sigma(s_1 x_1 + \cdots + s_n x_n),$$

onde  $s_i \in \{-1, 1\}$ , para cada  $i = 1, \dots, k$ , e a soma é tomada sobre todas as  $2^n$  configurações possíveis de  $s_1 \cdots s_n$ .

Além disso, essa é a menor rede de 1 camada capaz de fazer tal aproximação.



Dado  $x = (x_1, x_2, \dots, x_n)$ , uma rede  $N$  de 1 camada escondida é da forma

$$N(x) = \sum_{j=1}^m w_j \sigma \left( \sum_{i=1}^n a_{ij} x_i \right)$$

Dado  $x = (x_1, x_2, \dots, x_n)$ , uma rede  $N$  de 1 camada escondida é da forma

$$N(x) = \sum_{j=1}^m w_j \sigma \left( \sum_{i=1}^n a_{ij} x_i \right)$$

Queremos uma rede de tamanho  $m$  com pesos  $w_j$  e  $a_{ij}$  tal que, denotando  $\sigma_k = \sigma^{(k)}(0)$ :

$$\sigma_n \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^n = \prod_{i=1}^n x_i, \quad (2)$$

$$\sigma_k \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^k = 0, \forall k \in \{1, \dots, n-1\} \quad (3)$$

Dado  $x = (x_1, x_2, \dots, x_n)$ , uma rede  $N$  de 1 camada escondida é da forma

$$N(x) = \sum_{j=1}^m w_j \sigma \left( \sum_{i=1}^n a_{ij} x_i \right)$$

Queremos uma rede de tamanho  $m$  com pesos  $w_j$  e  $a_{ij}$  tal que, denotando  $\sigma_k = \sigma^{(k)}(0)$ :

$$\sigma_n \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^n = \prod_{i=1}^n x_i, \quad (2)$$

$$\sigma_k \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^k = 0, \forall k \in \{1, \dots, n-1\} \quad (3)$$

Mostremos que  $m = 2^n$  é necessário e suficiente.

$2^n$  **Suficiente** Sejam  $S_1, S_2, \dots, S_m$  subconjuntos de  $\{1, \dots, n\}$ .  
Defina p/ cada  $i \in \{1, \dots, n\}$

$$s_i(S) = \begin{cases} -1, i \in S, \\ 1, i \notin S. \end{cases}$$

$2^n$  **Suficiente** Sejam  $S_1, S_2, \dots, S_m$  subconjuntos de  $\{1, \dots, n\}$ .  
Defina p/ cada  $i \in \{1, \dots, n\}$

$$s_i(S) = \begin{cases} -1, i \in S, \\ 1, i \notin S. \end{cases}$$

Defina  $a_{ij} = s_i(S_j)$

$$w_j = x \frac{1}{2^n n! \sigma_n} \prod_{i=1}^n a_{ij} = \frac{(-1)^{|S_j|}}{2^n n! \sigma_n}.$$

Considere  $p(x) = x_1^{r_1} x_2^{r_2} \cdots x_n^{r_n}$ ,  $r_1 + \cdots + r_n = r \leq n$ . Vamos verificar que no desenvolvimento de

$$\sigma_r \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^r$$

se  $p(x) \neq \prod_i^n x_i$ , seu coeficiente é 0.

Se  $p(x) \neq \prod_i^n x_i$ , existe  $r_{i_0} = 0$ .

$$\sigma_r \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^r \quad (4)$$

$$= \sigma_r \sum_{j=1}^m \frac{(-1)^{|S_j|}}{2^n n! \sigma_n} \left( \sum_{i=1}^n s_i(S_j) x_i \right)^r \quad (5)$$

Se  $p(x) \neq \prod_i^n x_i$ , existe  $r_{i_0} = 0$ .

$$\sigma_r \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^r \quad (4)$$

$$= \sigma_r \sum_{j=1}^m \frac{(-1)^{|S_j|}}{2^n n! \sigma_n} \left( \sum_{i=1}^n s_i(S_j) x_i \right)^r \quad (5)$$

$$= \sigma_r \sum_{S_j \not\ni i_0}^m \left[ \frac{(-1)^{|S_j|}}{2^n n! \sigma_n} \left( \sum_{i=1}^n s_i(S_j) x_i \right)^r + \frac{(-1)^{|S_j \cup \{i_0\}|}}{2^n n! \sigma_n} \left( \sum_{i=1}^n s_i(S_j \cup \{i_0\}) x_i \right)^r \right]$$



Se  $p(x) \neq \prod_i^n x_i$ , existe  $r_{i_0} = 0$ .

$$\sigma_r \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^r \quad (4)$$

$$= \sigma_r \sum_{j=1}^m \frac{(-1)^{|S_j|}}{2^n n! \sigma_n} \left( \sum_{i=1}^n s_i(S_j) x_i \right)^r \quad (5)$$

$$= \sigma_r \sum_{S_j \not\ni i_0}^m \left[ \frac{(-1)^{|S_j|}}{2^n n! \sigma_n} \left( \sum_{i=1}^n s_i(S_j) x_i \right)^r + \frac{(-1)^{|S_j \cup \{i_0\}|}}{2^n n! \sigma_n} \left( \sum_{i=1}^n s_i(S_j \cup \{i_0\}) x_i \right)^r \right]$$

$$= \sigma_r \frac{(-1)^{|S_j|}}{2^n n! \sigma_n} \sum_{S_j \not\ni i_0}^m \left[ \left( \sum_{i=1}^n s_i(S_j) x_i \right)^r - \left( \sum_{i=1}^n s_i(S_j \cup \{i_0\}) x_i \right)^r \right]$$

Veja que  $s_i(S_j) = s_i(S_j \cup \{i_0\})$ ,  $\forall i \neq i_0$ . Logo, se  $r_{i_0} = 0$  em  $p(x)$ , os coeficientes são iguais, portanto, se cancelam em

$$\left( \sum_{i=1}^n s_i(S_j) x_i \right)^r - \left( \sum_{i=1}^n s_i(S_j \cup \{i_0\}) x_i \right)^r$$

Veja que  $s_i(S_j) = s_i(S_j \cup \{i_0\})$ ,  $\forall i \neq i_0$ . Logo, se  $r_{i_0} = 0$  em  $p(x)$ , os coeficientes são iguais, portanto, se cancelam em

$$\left( \sum_{i=1}^n s_i(S_j) x_i \right)^r - \left( \sum_{i=1}^n s_i(S_j \cup \{i_0\}) x_i \right)^r$$

Isso vale para cada  $i_0$ . Em particular, para cada  $r < n$ , vale que:

$$\sigma_r \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^k = 0,$$

como queríamos;

Se  $p(x) = \prod x_i$ , o coeficiente na expansão de  $(\sum a_{ij}x_i)^n$  é

$$n! \prod_i a_{ij} = n!(-1)^{|S_j|}$$

Se  $p(x) = \prod x_i$ , o coeficiente na expansão de  $(\sum a_{ij}x_i)^n$  é

$$n! \prod_i a_{ij} = n!(-1)^{|S_j|}$$

Logo,

$$\begin{aligned} & \sigma_n \sum_j^m w_j \left( \sum a_{ij}x_i \right)^n \\ &= \sigma_n \sum_j^{2^n} \frac{(-1)^{|S_j|}}{2^n n! \sigma_n} \left( \sum a_{ij}x_i \right)^n \\ &= \sigma_n \sum_j^{2^n} \frac{(-1)^{|S_j|}}{2^n n! \sigma_n} n!(-1)^{|S_j|} = \prod_{i=1}^n x_i \end{aligned}$$

como queríamos.

$2^n$  é **Necessário**: Suponha que existe uma rede de uma camada escondida com  $m$  neurônios e pesos  $w_j, a_{ij}$  que satisfaz as condições desejadas:

$$\sigma_n \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^n = \prod_{i=1}^n x_i,$$

$$\sigma_k \sum_{j=1}^m w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^k = 0, \forall k \in \{1, \dots, n-1\}$$

Mostremos que  $m \geq 2^n$ .

Seja  $S \subset \{1, \dots, n\}$ . Tomando todas as parciais  $\frac{\partial}{\partial x_h}$  para  $h \in S$  nas duas equações:

$$\frac{n! \sigma_n}{|n - S|!} \sum_{j=1}^m w_j \prod_{h \in S} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-|S|} = \prod_{h \notin S} x_h, \quad (6)$$

$$\frac{k! \sigma_k}{|k - S|!} \sum_{j=1}^m w_j \prod_{h \in S} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{k-|S|} = 0, k \geq |S| \quad (7)$$

Sejam  $S_1, \dots, S_{2^n}$  os subconjuntos de  $\{1, \dots, n\}$  e defina  $A \in \mathbb{R}^{2^n \times m}$  por

$$A_{ij} = \prod_{h \in S_i} a_{hj}.$$

Ideia: mostrar que  $A$  tem posto linha completo.



Sejam  $S_1, \dots, S_{2^n}$  os subconjuntos de  $\{1, \dots, n\}$  e defina  $A \in \mathbb{R}^{2^n \times m}$  por

$$A_{ij} = \prod_{h \in S_i} a_{hj}.$$

Ideia: mostrar que  $A$  tem posto linha completo.

Suponha por contradição que exista uma dependência linear nas linhas de  $A$ :

$$c^T A = \sum_l^r c_l A_l = 0$$

com cada  $S_l$  distinto entre si e  $c_l \neq 0$ , para cada  $l$ . Seja  $s = \max_{\ell | \sum_l^r c_l A_l = 0} |S_\ell|$ .

Sejam  $S_1, \dots, S_{2^n}$  os subconjuntos de  $\{1, \dots, n\}$  e defina  $A \in \mathbb{R}^{2^n \times m}$  por

$$A_{ij} = \prod_{h \in S_i} a_{hj}.$$

Ideia: mostrar que  $A$  tem posto linha completo.

Suponha por contradição que exista uma dependência linear nas linhas de  $A$ :

$$c^T A = \sum_l^r c_l A_l = 0$$

com cada  $S_l$  distinto entre si e  $c_l \neq 0$ , para cada  $l$ . Seja  $s = \max_l |\sum_l^r c_l A_l = 0| S_l|$ . Defina  $d \in \mathbb{R}^m$  por

$$d_j = w_j \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-s}.$$

Então,

$$0 = \mathbf{c}^t \mathbf{A} \mathbf{d} = \sum_{\ell=1}^r c_{\ell} \sum_{j=1}^m w_j \prod_{h \in S_{\ell}} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-s}$$

Então,

$$\begin{aligned}
 0 &= \mathbf{c}^t \mathbf{A} \mathbf{d} = \sum_{\ell=1}^r c_{\ell} \sum_{j=1}^m w_j \prod_{h \in S_{\ell}} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-s} \\
 &= \sum_{\ell \mid (|S_{\ell}|=s)} c_{\ell} \sum_{j=1}^m w_j \prod_{h \in S_{\ell}} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-|S_{\ell}|}
 \end{aligned}$$

Então,

$$\begin{aligned}
 0 &= \mathbf{c}^t \mathbf{A} \mathbf{d} = \sum_{\ell=1}^r c_{\ell} \sum_{j=1}^m w_j \prod_{h \in S_{\ell}} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-s} \\
 &= \sum_{\ell \mid (|S_{\ell}|=s)} c_{\ell} \sum_{j=1}^m w_j \prod_{h \in S_{\ell}} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-|S_{\ell}|} \\
 &+ \sum_{\ell \mid (|S_{\ell}|<s)} c_{\ell} \sum_{j=1}^m w_j \prod_{h \in S_{\ell}} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{(n+|S_{\ell}|-s)-|S_{\ell}|} .
 \end{aligned}$$

Isto é,

$$\begin{aligned}
 0 = & \sum_{\ell \mid (|S_\ell|=s)} c_\ell \sum_{j=1}^m w_j \prod_{h \in S_\ell} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-|S_\ell|} \\
 + & \sum_{\ell \mid (|S_\ell|<s)} c_\ell \sum_{j=1}^m w_j \prod_{h \in S_\ell} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{(n+|S_\ell|-s)-|S_\ell|} .
 \end{aligned}$$

Isto é,

$$\begin{aligned} 0 = & \sum_{\ell \mid (|S_\ell|=s)} c_\ell \sum_{j=1}^m w_j \prod_{h \in S_\ell} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-|S_\ell|} \\ & + \sum_{\ell \mid (|S_\ell|<s)} c_\ell \sum_{j=1}^m w_j \prod_{h \in S_\ell} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{(n+|S_\ell|-s)-|S_\ell|}. \end{aligned}$$

Agora, aplicando (7), i.e., com  $k = (n + |S_\ell| - s) - |S_\ell|$ , temos que a segunda parte da soma é igual a 0.

Por outro lado, substituindo (6) acima, temos

$$\begin{aligned} 0 &= \sum_{\ell \mid (|S_\ell|=s)} c_\ell \sum_{j=1}^m w_j \prod_{h \in S_\ell} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-|S_\ell|} \\ &= \sum_{\ell \mid (|S_\ell|=s)} c_\ell \frac{|n - S_\ell|!}{n! \sigma_n} \prod_{h \notin S_\ell} x_h \end{aligned}$$



Por outro lado, substituindo (6) acima, temos

$$\begin{aligned} 0 &= \sum_{\ell \mid (|S_\ell|=s)} c_\ell \sum_{j=1}^m w_j \prod_{h \in S_\ell} a_{hj} \left( \sum_{i=1}^n a_{ij} x_i \right)^{n-|S_\ell|} \\ &= \sum_{\ell \mid (|S_\ell|=s)} c_\ell \frac{|n - S_\ell|!}{n! \sigma_n} \prod_{h \notin S_\ell} x_h \end{aligned}$$

Ou seja, temos uma soma não trivial de monômios linearmente independentes igual a zero. Portanto,  $A \in \mathbb{R}^{2^n \times m}$  tem posto linha cheio, e  $m \geq 2^n$ .  $\square$

O teorema ilustra como reduzir o número de camadas pode não ser eficiente.

**Exemplo:** para fazer o produto de  $n = 8$  variáveis distintas, uma rede de 1 camada escondida precisaria de  $2^8 = 256$  neurônios. Porém, podemos aproximar por uma rede de 3 camadas escondidas, totalizando 28 neurônios.

## Artigo: Why does deep and cheap learning work so well?

Introdução

Expressabilidade e Eficiência de Redes Rasas

Custos de Achatamento

Interpretações físicas

## Referências

Considere  $p : X \rightarrow (0, 1)$  uma distribuição de probabilidade sobre um espaço  $X$ . A probabilidade condicional:

$$p(\mathbf{x}|\mathbf{y}) = \frac{\mathbf{p}(\mathbf{x} \cup \mathbf{y})}{\mathbf{p}(\mathbf{y})} \quad (8)$$

- ▶  $\mathbf{x}$  varia no espaço amostral;
- ▶  $y$  é outra variável condicionando, ou um parâmetro do modelo

Considere  $p : X \rightarrow (0, 1)$  uma distribuição de probabilidade sobre um espaço  $X$ . A probabilidade condicional:

$$p(\mathbf{x}|\mathbf{y}) = \frac{\mathbf{p}(\mathbf{x} \cup \mathbf{y})}{\mathbf{p}(\mathbf{y})} \quad (8)$$

- ▶  $\mathbf{x}$  varia no espaço amostral;
- ▶  $y$  é outra variável condicionando, ou um parâmetro do modelo

## Exemplos:

- ▶  $\mathbf{x}$  é o vetor de pixels da imagem e  $y$  é um elemento de um conjunto de animais {Gato, Cachorro, Coelho, etc.};
- ▶  $\mathbf{x}$  é a magnetização em cada parte de uma barra de metal,  $y$  é um elemento de um conjunto de elementos {Ferro, Alumínio, etc.}

Calcular  $p(\mathbf{x}|y)$  é conhecido como problema de **predição**, e  $p(y|\mathbf{x})$  de **classificação**.

Calcular  $p(\mathbf{x}|y)$  é conhecido como problema de **predição**, e  $p(y|\mathbf{x})$  de **classificação**.

Para calcular  $p(y|\mathbf{x})$ , usamos o **Teorema de Bayes**:

$$p(y|\mathbf{x}) = \frac{p(\mathbf{x}|y)p(y)}{\sum_{y' \in Y} p(\mathbf{x}|y')p(y')} \quad \left( = \frac{p(\mathbf{x}|y)p(y)}{p(\mathbf{x})} \right)$$

- $p(y)$  é chamada informação a priori;

Estamos interessados nas quantidades

- ▶  $H_y(\mathbf{x}) = -\ln p(\mathbf{x}|y)$ , chamado **Hamiltoniano**
- ▶  $\mu_y = -\ln p(y)$ .

**Exemplo:** se  $p(x) = \frac{e^{-x^2}}{\sqrt{2\pi}}$ , então

$$\mu_y = -\ln \left( \frac{e^{-x^2}}{\sqrt{2\pi}} \right) = x^2 - \ln(\sqrt{2\pi}).$$



Uma distribuição  $p(x|y)$  pode ser escrita de forma geral como

$$p(y|x) = \frac{1}{S(x)} e^{-[H_y(x) - \mu_x]}$$

com

$$S(x) = \sum_y e^{-[H_y(x) - \mu_x]}$$

Uma distribuição  $p(x|y)$  pode ser escrita de forma geral como

$$p(y|x) = \frac{1}{S(x)} e^{-[H_y(x) - \mu_x]}$$

com

$$S(x) = \sum_y e^{-[H_y(x) - \mu_x]}$$

Ou, de forma vetorial, juntando os Hamiltonianos de cada variável  $p_y$  em vetores:

$$\mathbf{p}(\mathbf{x}) = \frac{1}{S(\mathbf{x})} e^{-[\mathbf{H}(\mathbf{x}) - \mu]}$$

**Motivação:** Se utilizarmos como função de ativação a Softmax

$$\sigma_i(\mathbf{x}) = \frac{1}{\sum_{j=1}^n e^{x_j}} (e^{x_1}, \dots, e^{x_n})$$

juntando cada um dos  
podemos escrever a equação anterior como:

$$\mathbf{p}(\mathbf{x}) = \sigma(-\mathbf{H}(\mathbf{x}) - \mu)$$

**Motivação:** Os autores atestam que os fenômenos físicos, em geral, são descritos por Hamiltonianos relativamente simples:

**Motivação:** Os autores atestam que os fenômenos físicos, em geral, são descritos por Hamiltonianos relativamente simples: polinômios ou podem ser aproximados por séries de Taylor.

**Motivação:** Os autores atestam que os fenômenos físicos, em geral, são descritos por Hamiltonianos relativamente simples: polinômios ou podem ser aproximados por séries de Taylor. E.g.

$$H_y(x) = \sum_x^d h + \sum h_i x_i + \sum_{i \leq j} h_{ij} x_i x_j + \sum_{i \leq j \leq k} h_{ijk} x_i x_j x_k + \cdots$$

Alguns motivos levantados pelos autores são:

Alguns motivos levantados pelos autores são:

1. Hamiltonianos da física tem ordem polinomial baixa (de 2 a 4 em geral);



Alguns motivos levantados pelos autores são:

1. Hamiltonianos da física tem ordem polinomial baixa (de 2 a 4 em geral);
2. Diversos fenômenos são *locais*, acontecem numa vizinhança muito próxima;
3. Muitos fenômenos apresentam simetrias, que diminuem drasticamente o número de parâmetros.

1. Hierarquia de processos (estocásticos, probabilísticos, etc);

1. Hierarquia de processos (estocásticos, probabilísticos, etc);
2. Os processos que controlam cada etapa são modelados por poucos parâmetros;

1. Hierarquia de processos (estocásticos, probabilísticos, etc);
2. Os processos que controlam cada etapa são modelados por poucos parâmetros;
3. Certas funções são caras para achatar, sendo mais valioso aproximar com mais camadas;

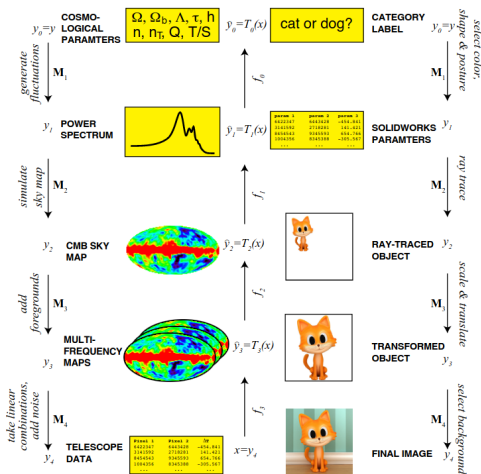


Figura: Diagrama ilustrando a cadeia de processos físicos

Artigo: Why does deep and cheap learning work so well?

Introdução

Expressabilidade e Eficiência de Redes Rasas

Custos de Achatamento

Interpretações físicas

Referências



Obrigado!

Contato: riffel.felipe@grad.ufsc.br

Repositório com os materiais da apresentação:  
<https://github.com/felipekriffel/Mestrado>