



Não Linearidade de Features e Parâmetros

Por Emanuel Lopes Silva e Matheus Costa Alves

Sumário

PRINCIPAIS TÓPICOS DISCUTIDOS
NESTA APRESENTAÇÃO

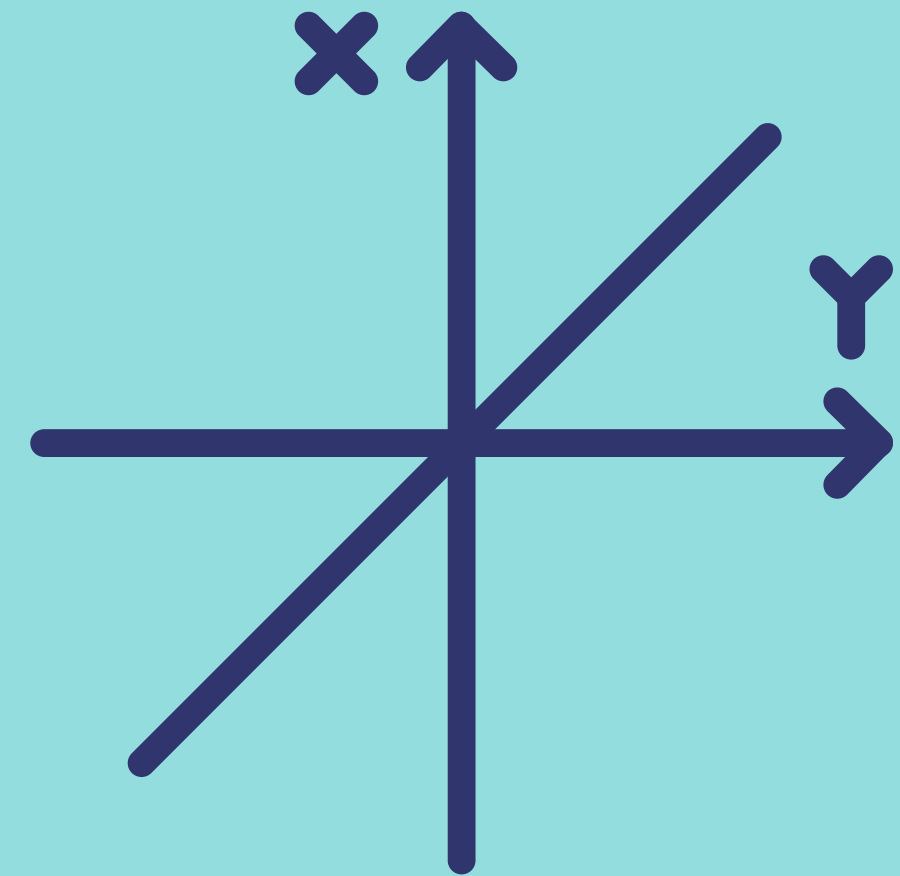
- Conceitos Gerais
 - Definição Geral.....4
 - Linear vs Não Linear.....5
- Não Linearidade de Features.....7
 - Limitações dos Modelos Lineares.....11
 - Engenharia de Features.....14
 - Interações entre Variáveis.....22
- Não Linearidade de Parâmetros.....26
 - Modelos Com Parâmetros Não Lineares.....29
 - Consequências da Não Linearidade dos Parâmetros.....34
 - Redes Neurais.....39
 - Funções de Ativação.....43
 - Teorema da Aproximação Universal.....49
 - Limitações e Implicações.....52

Conceitos Gerais



Definição

A não linearidade representa uma situação em que a relação entre os elementos de um sistema não é proporcional, direta ou simples





Não Linear

- A relação entre variáveis não é proporcional.
- Gráfico: curvas, parábolas, ondas, etc.



Linear

- Uma relação proporcional entre variáveis.
- Gráfico: sempre uma reta (em 2D) ou plano (em 3D).



Não Linear

- Mais flexíveis: podem modelar padrões complexos.
- Não seguem uma única inclinação.
- Pequenas mudanças em x podem causar grandes mudanças em y .

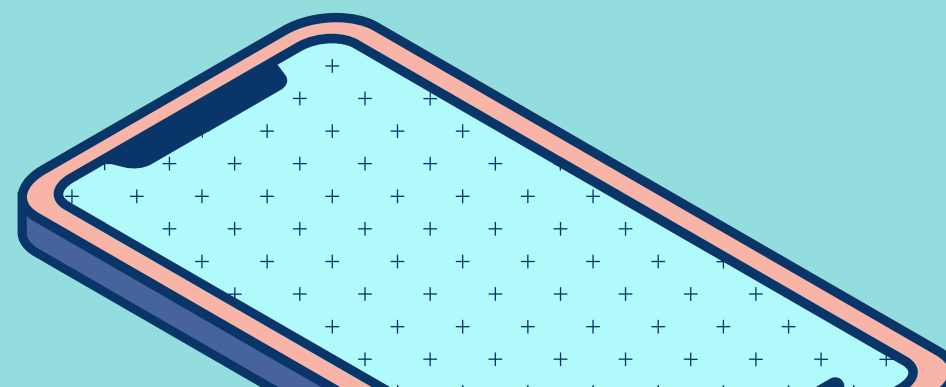


Linear

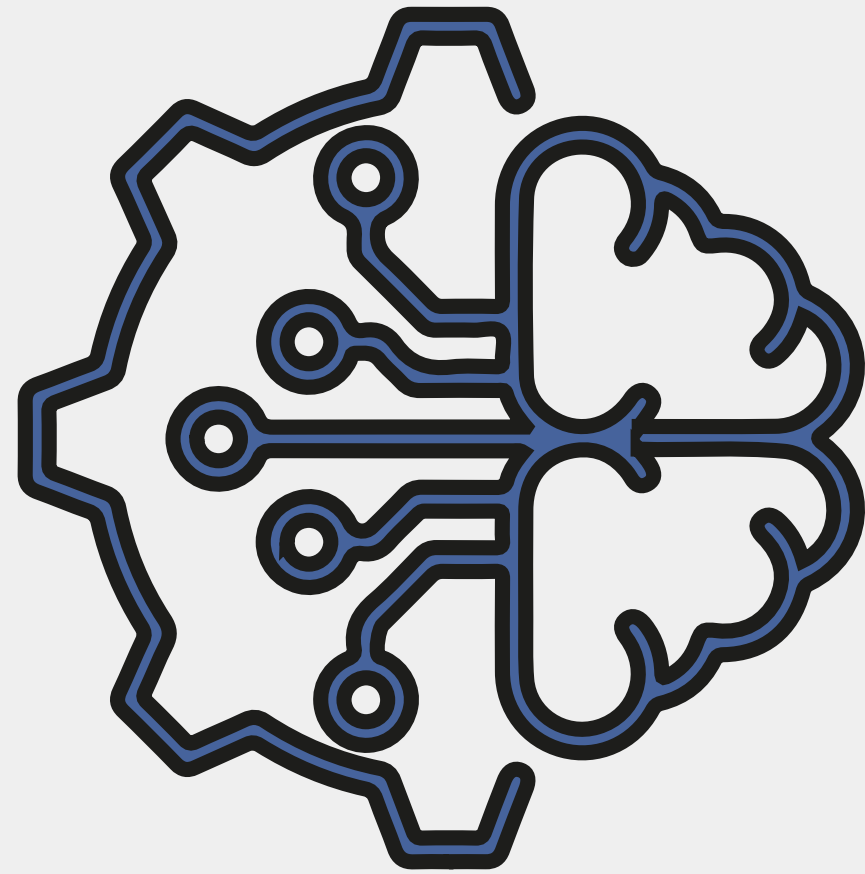
- Simples de interpretar.
- Coeficiente angular (a) e linear (b) definem a relação
- Proporcionalidade constante: Dobrar x dobra y .



Não Linearidade de Features



Não Linearidade das Features



Não linearidade de features significa que os dados de entrada apresentam padrões complexos que não podem ser separados ou modelados por uma função linear simples

- As interações entre variáveis podem ser curvas, dobradas ou altamente interdependentes
- Os dados do mundo real não seguem relações diretas do tipo "aumenta um pouco aqui, aumenta um pouco ali"

Por Que a Não Linearidade de Features é Necessária?

1 ————— 2 ————— 3 ————— 4 ————— 5

Motivo

Padrões em imagens, sons, textos e dados financeiros não podem ser separados ou explicados apenas por linhas retas ou planos.

Motivo

Um modelo linear só consegue aprender relações proporcionais e diretas

Motivo

Não Linearidade Ajuda a chegar Generalização para Novos Casos

Motivo

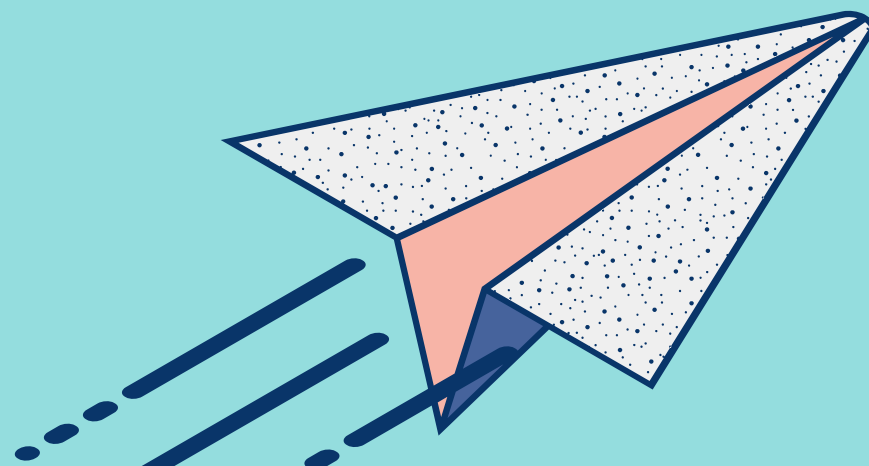
Permite o aprendizado de padrões complexos.

Motivo

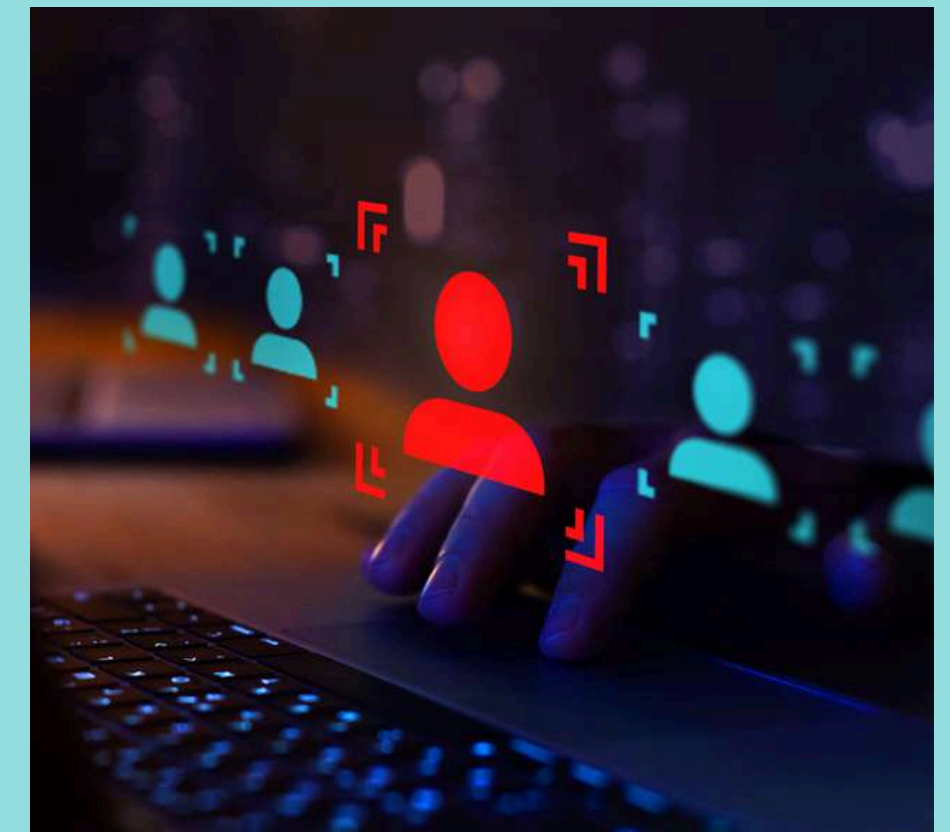
Dados Reais, na maioria das vezes, têm interações complexas entre variáveis

Importância de Modelar Padrões Complexos

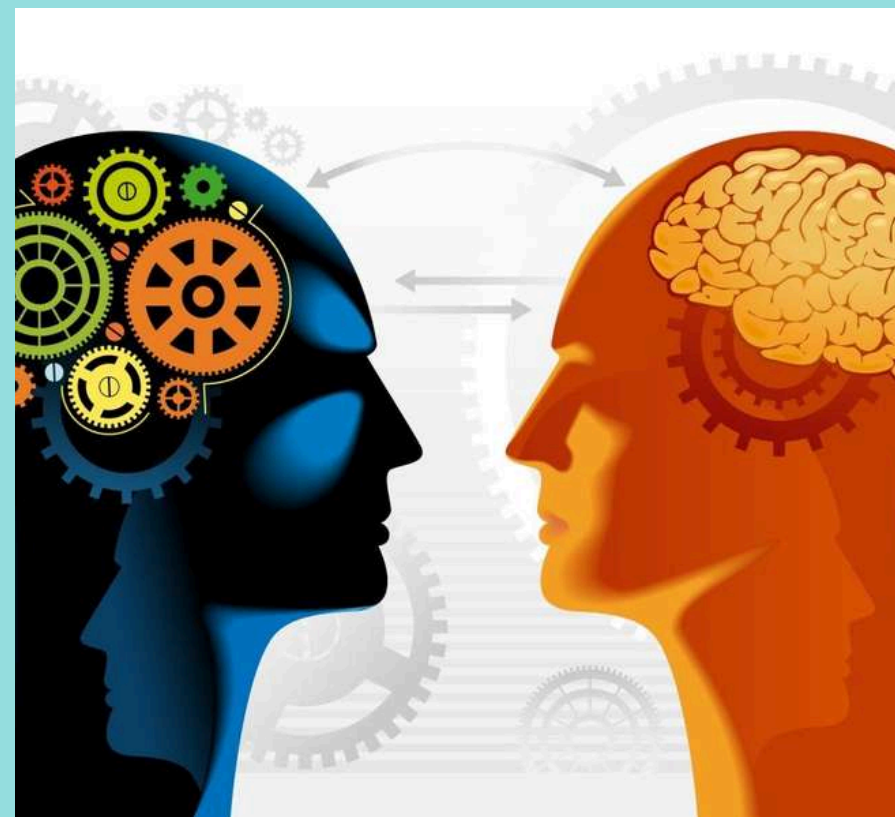
- A Maior Parte do Mundo Real Não É Linear
- Modelos Lineares São Limitados
- Flexibilidade e Capacidade de Representação



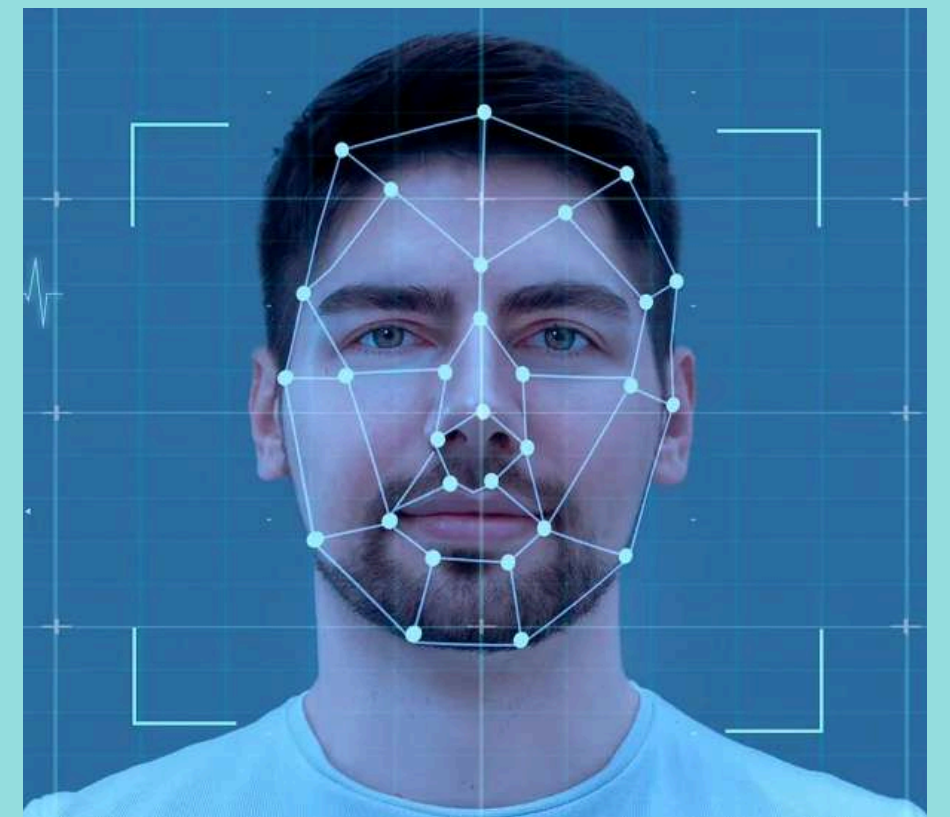
Diagnóstico Médico



Detecção de Fraudes

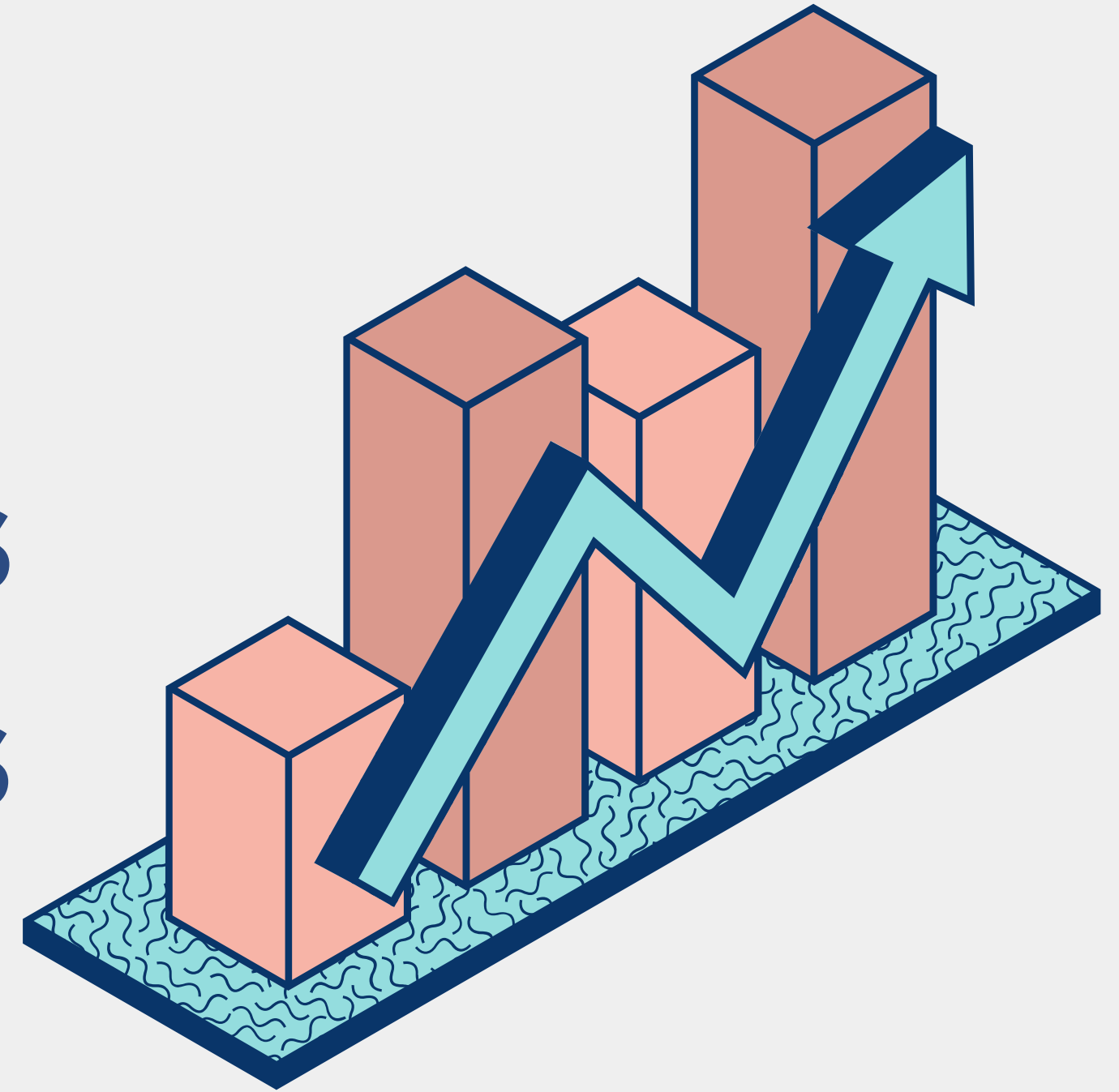


Processamento de Linguagem



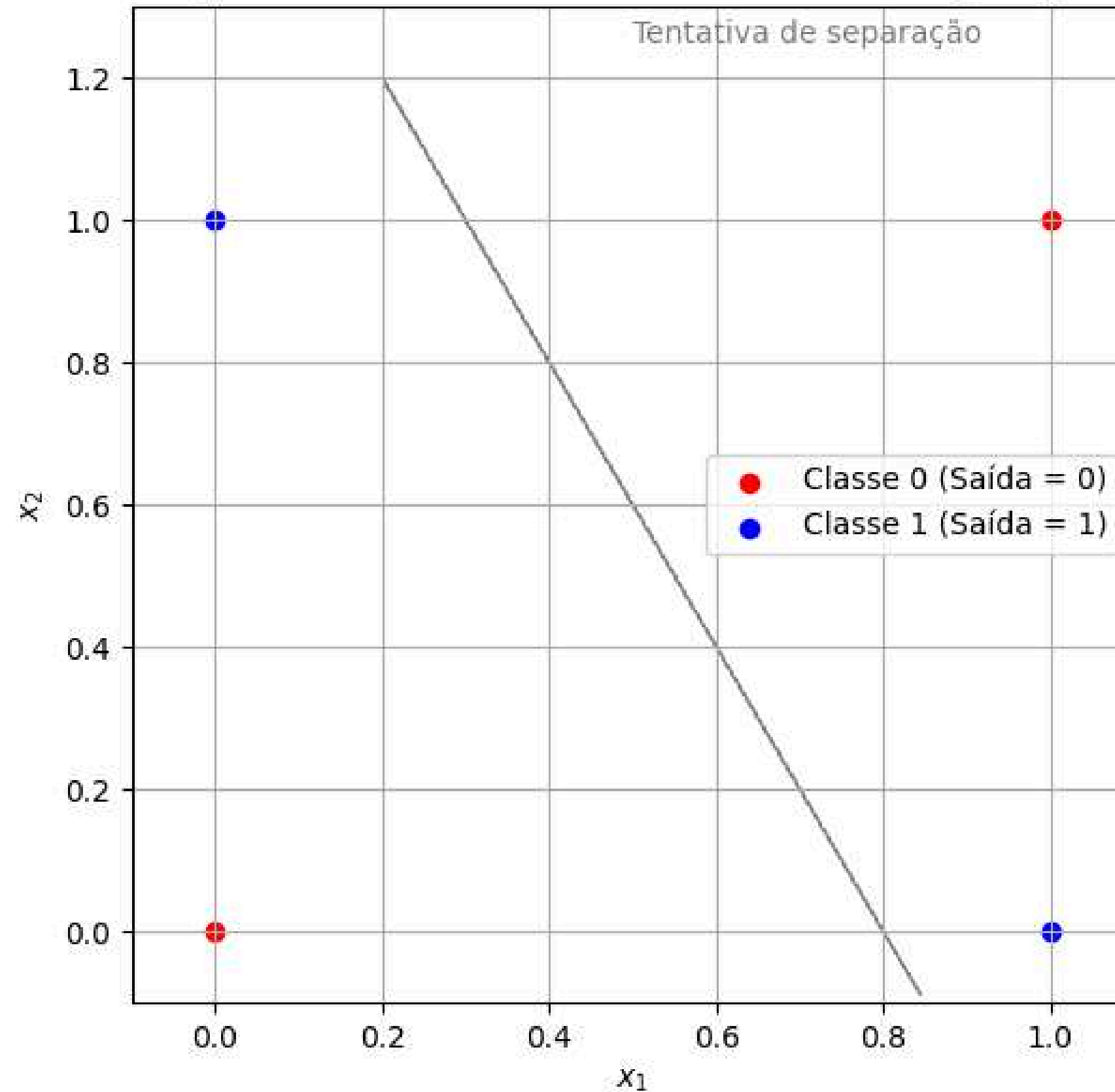
Reconhecimento de Imagens

Limitações de Modelos Lineares Frente aos Dados



Exemplo: XOR

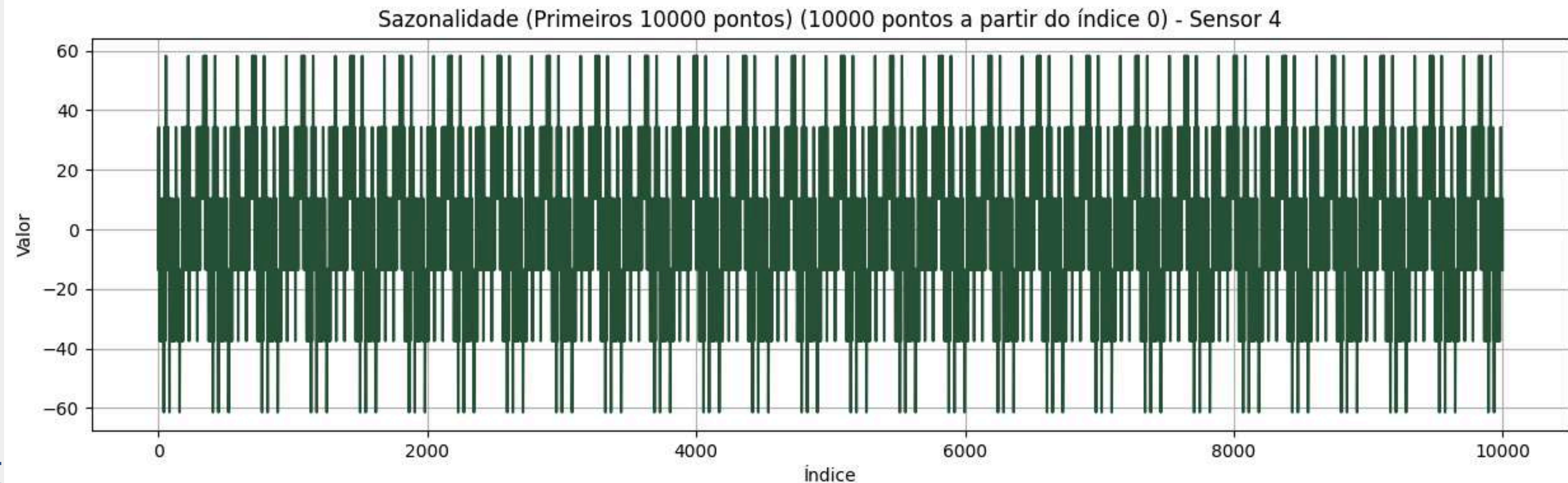
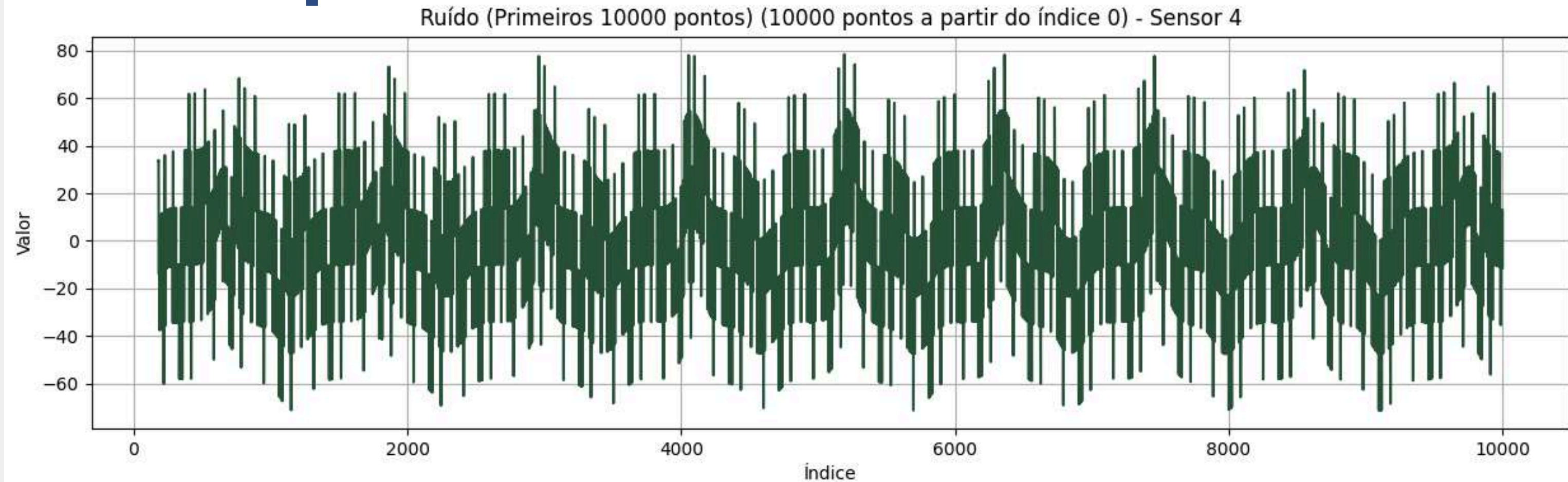
Visualização do Problema XOR com Tentativa de Separação Linear



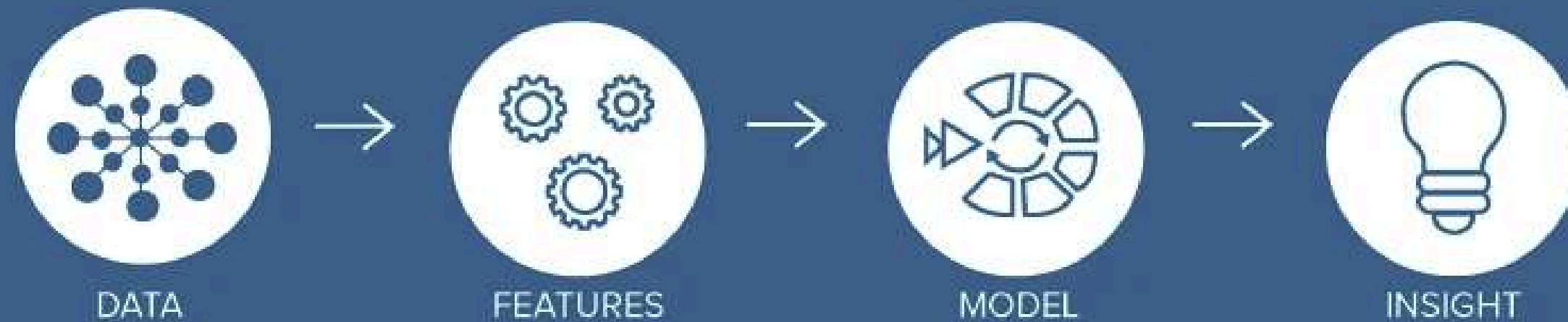
Motivo

- Não há reta que separe as classes.

Exemplo: Sazonalidade e Ruído

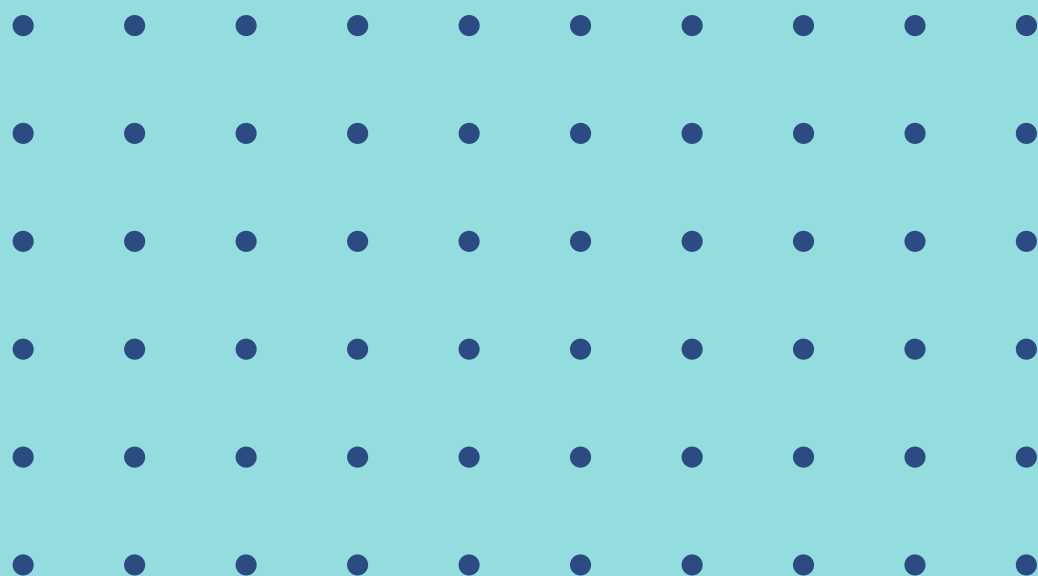
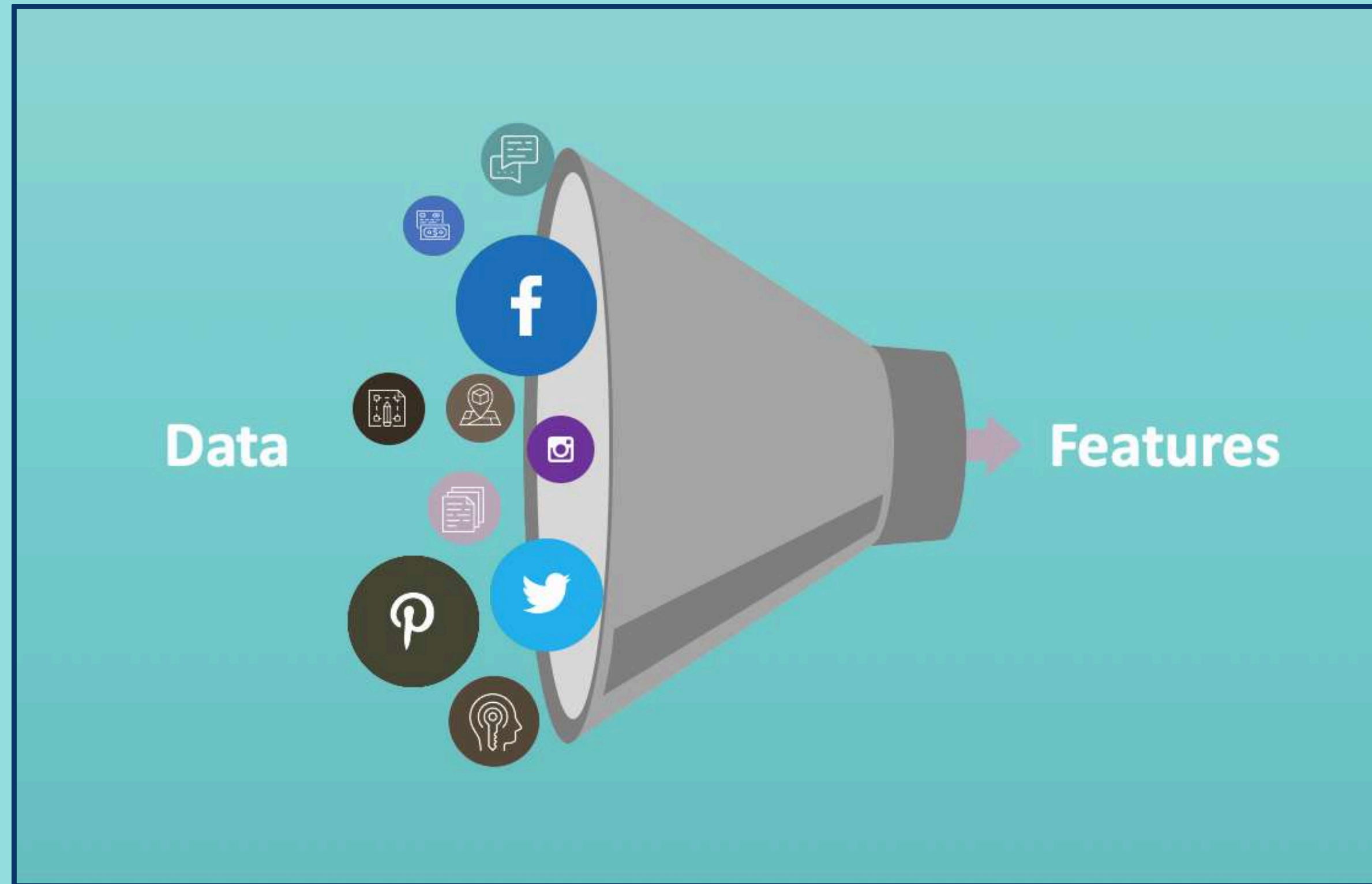


Importância da Engenharia de Features



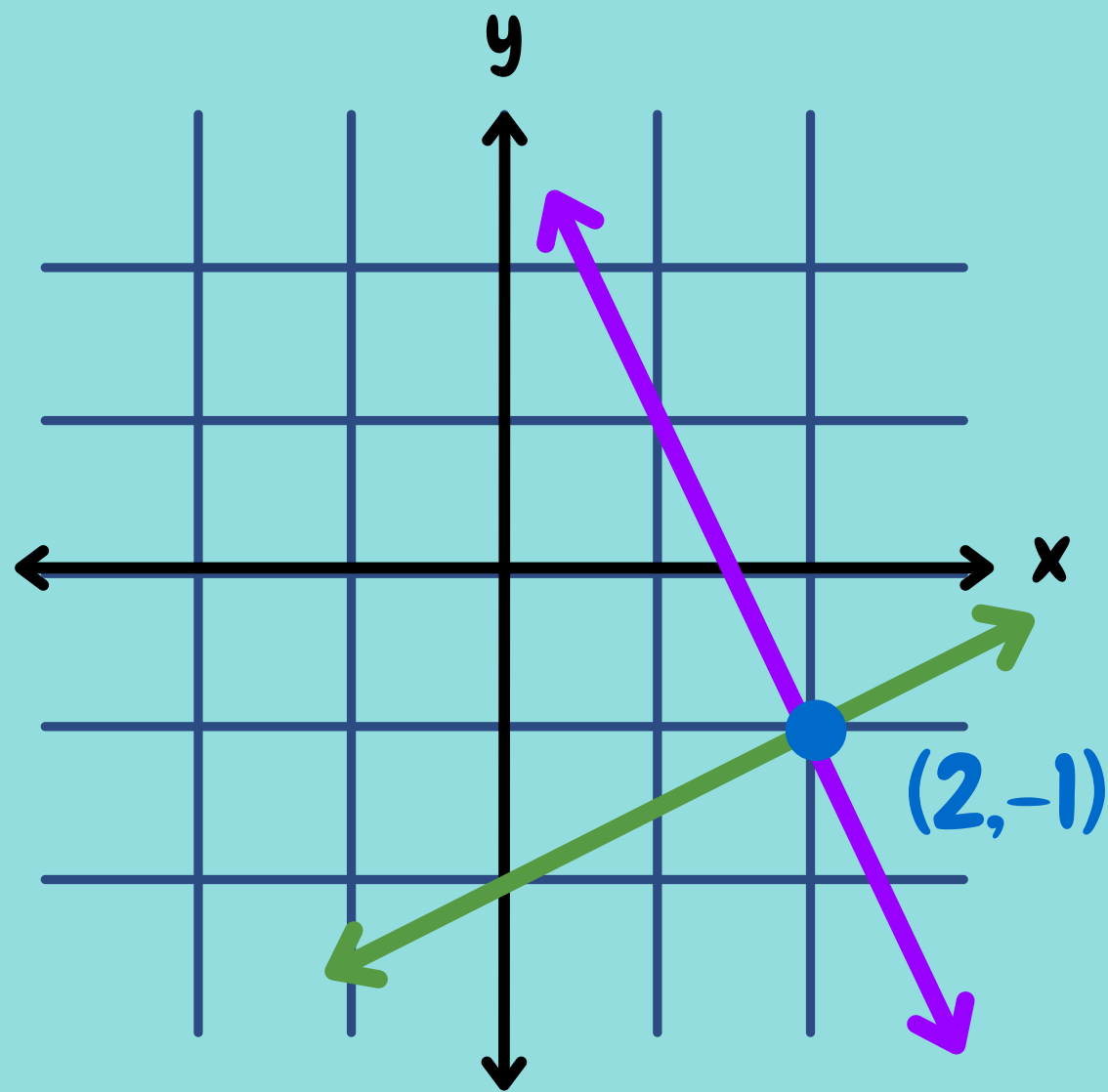
O que é Engenharia de Features?

- Criar, combinar e editar features
- Auxilia o modelo a enxergar padrões complexos com uma representação mais adequada



Por Que Fazer Isso?

Como “Linearizar” o Problema



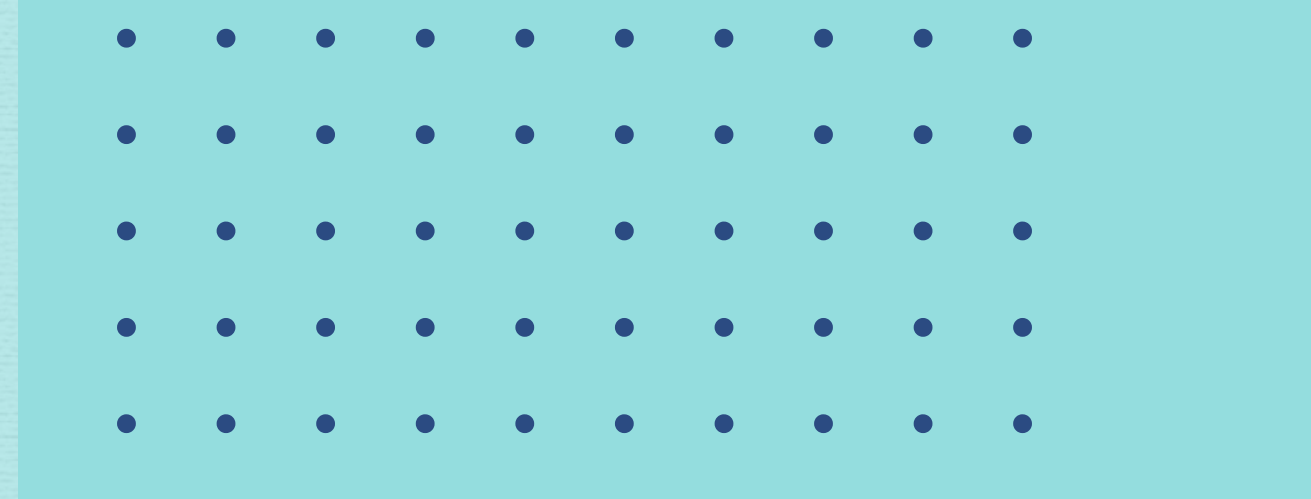
Muitas vezes, os dados não são linearmente separáveis em sua forma original

Dessa forma, temos que adaptar para o modelo poder entender

Pode-se criar uma nova feature baseada em uma combinação não linear das variáveis

Criar novas features ajuda a expor padrões escondidos no espaço original

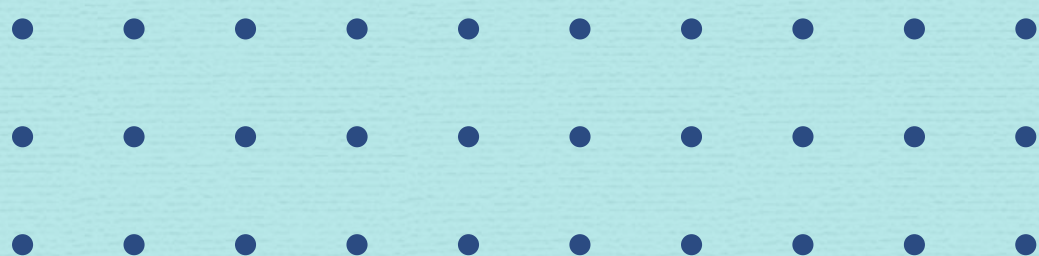
Exemplo: XOR



01 Deve-se criar uma Nova Feature

A ideia é adicionar uma nova dimensão ao problema, para desenrolar a complexidade

- $Z = X1 \times X2$
- Agora, o problema tem três dimensões: $X1$, $X2$ e Z



02 O que Representa?

Z é uma feature de interação:

- Quando $X1=0$ ou $X2 = 0$, $Z= 0 \rightarrow$ Nenhuma interação ocorre
- Quando $X1 = 1$ e $X2 =1$, $Z=1 \rightarrow$ Existe uma interação única que diferencia o ponto

Exemplo: XOR

03 Como isso Resolve o Problema?

Agora o modelo pode usar Z como base para uma decisão linear simples:

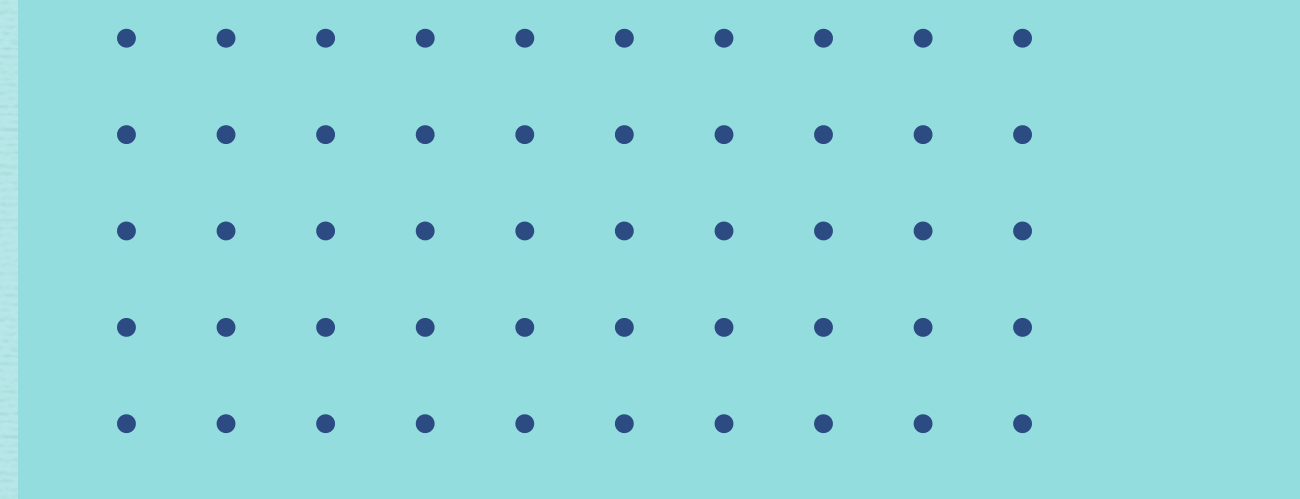
- Quando $Z=1$, a classe é 0
- Quando $Z=0$, a classe é 1
- É a tentativa de capturar uma relação não linear escondida nos dados originais $X1$ e $X2$



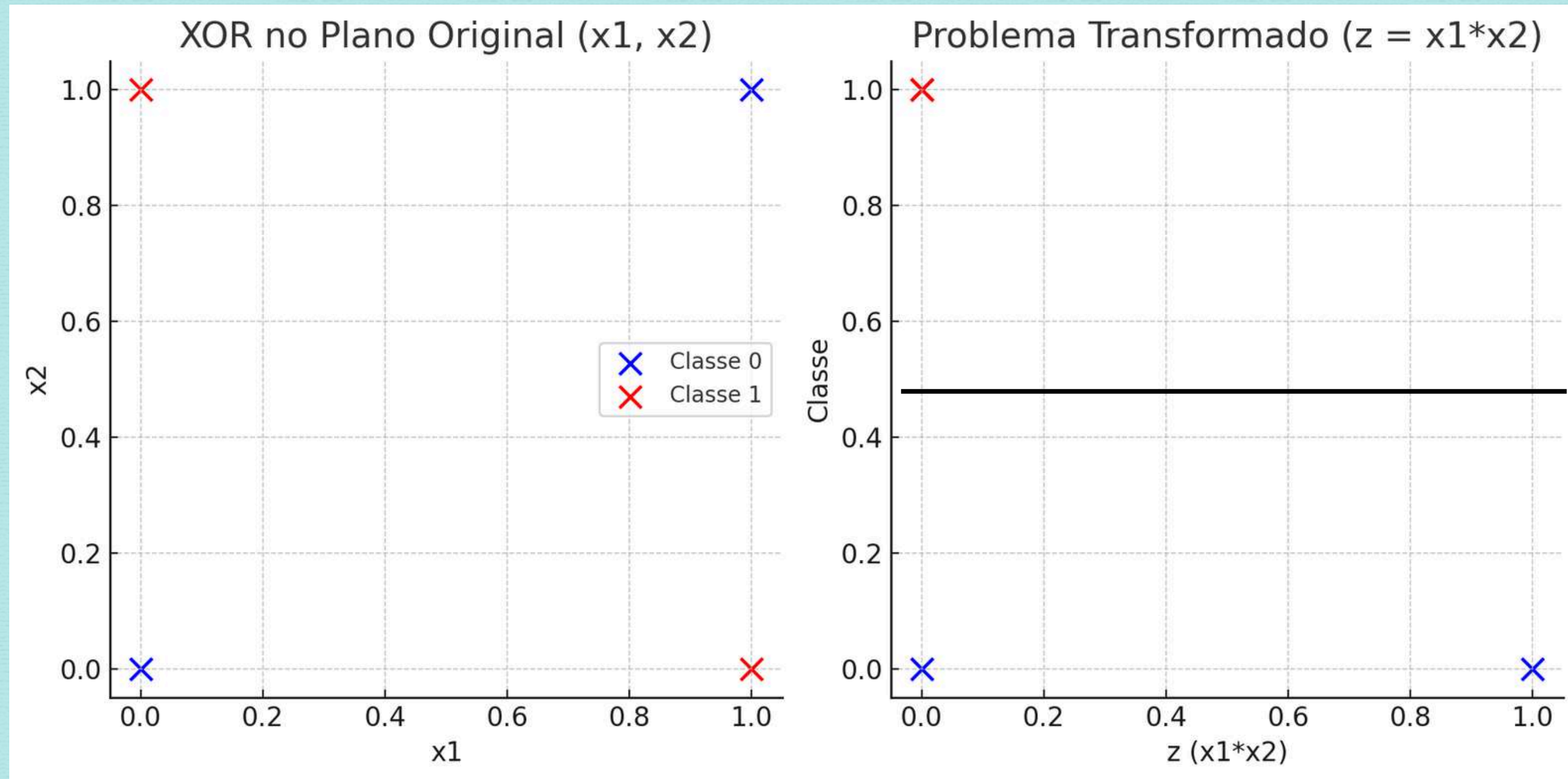
04 Intuição Geométrica

No plano $X1$ e $X2$, os pontos estão embaralhados, não dá para traçar uma linha reta que separe as classes

Quando usamos $Z=X1 \times X2$, projetamos os dados em um eixo onde a separação linear é possível, ao adicionar uma nova dimensão

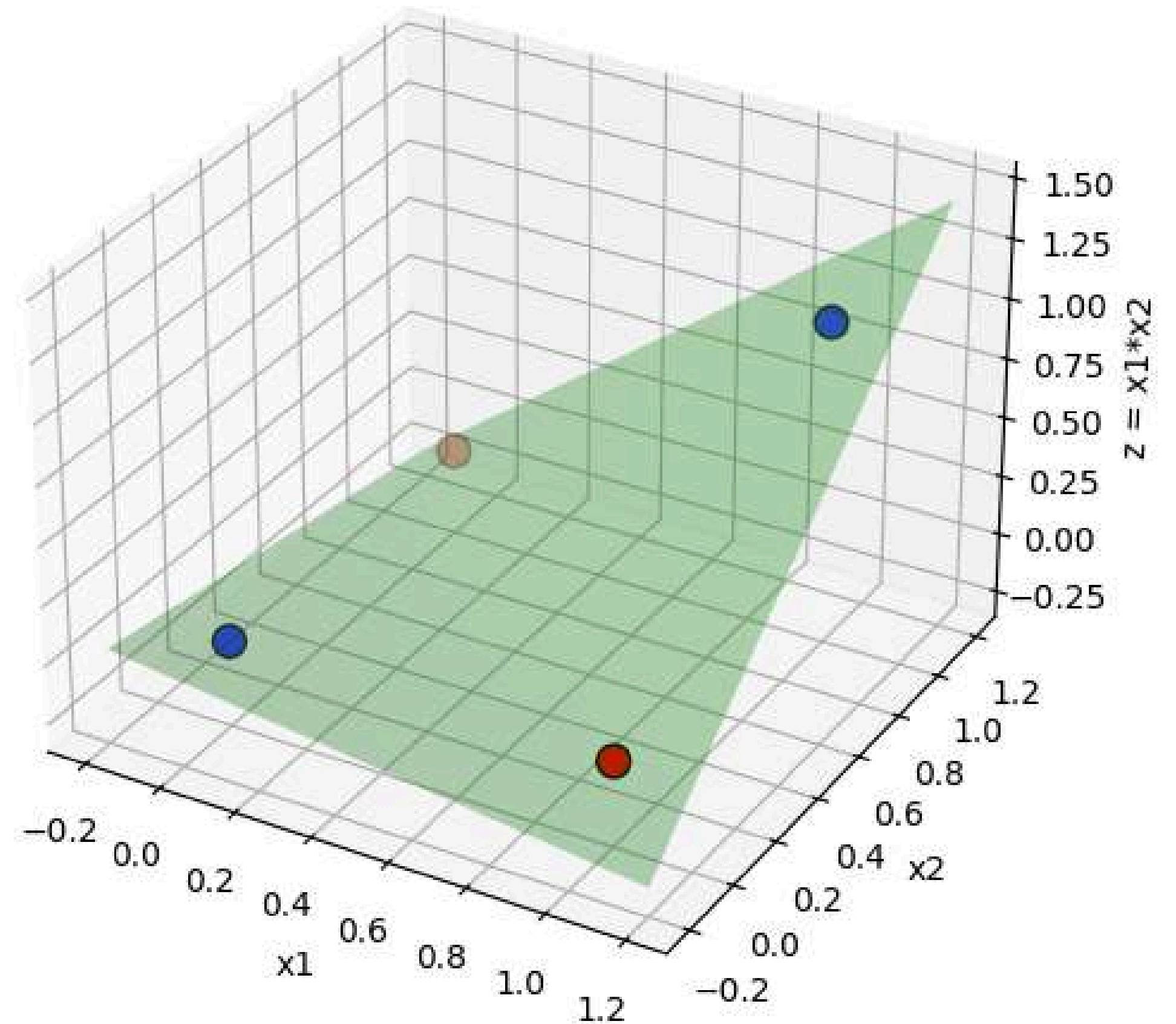


Exemplo: XOR



Exemplo: XOR

Plano Transformado: $(x_1, x_2, z=x_1*x_2)$



Limitações da Engenharia de Features

Conhecimento Especializado

É preciso entender o problema profundamente para saber quais variáveis combinar

Muitas Interações

Em imagens, áudio e texto, são necessários milhares ou milhões de interações complexas

Risco de Omitir Padrões Importantes

Sem a feature certa, o modelo linear não consegue aprender o padrão



Interações entre Variáveis

O que são interações entre variáveis?

Definição

Interação entre variáveis ocorre quando o efeito combinado de duas (ou mais) variáveis não pode ser explicado apenas pela soma de seus efeitos individuais

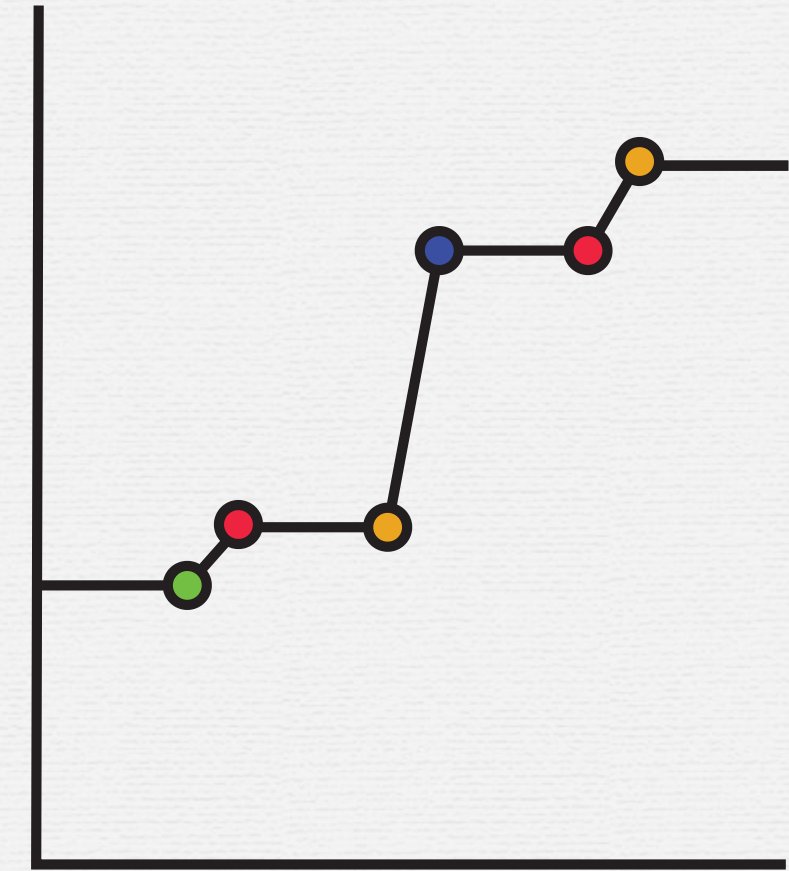




O que são interações entre variáveis?

Para que serve?

- Interações entre variáveis criam padrões complexos que não podem ser aprendidos por modelos lineares simples
- Capturar essas interações é o que permite entender fenômenos do mundo real, como doenças, imagens e comportamento humano



Exemplos

Diagnóstico Médico

Em problemas de saúde, a interação entre variáveis como altura, idade e dieta é fundamental

O risco de uma doença não depende apenas de cada fator isoladamente, mas do efeito combinado entre eles

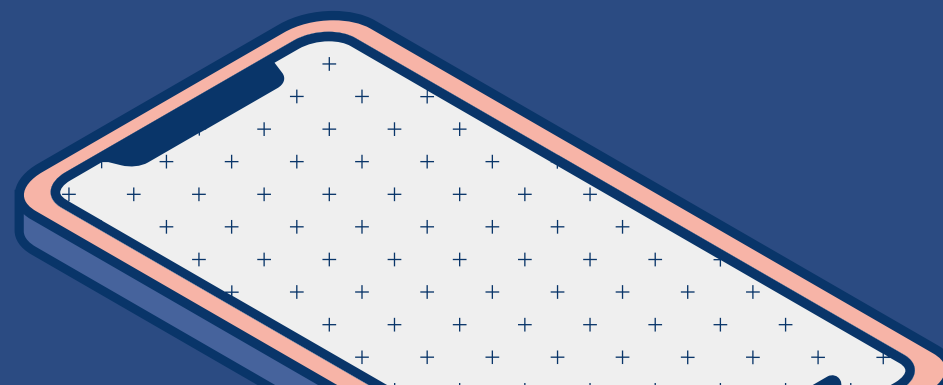
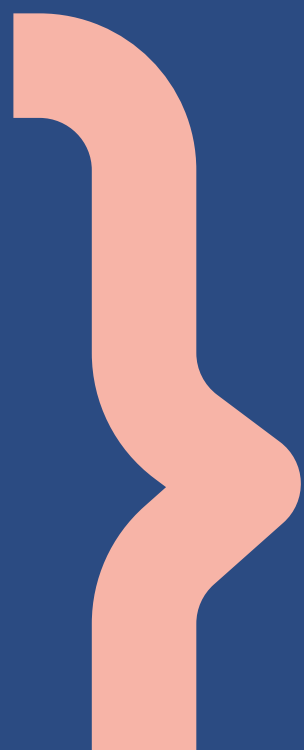
Reconhecimento de Dígitos

No reconhecimento de imagens, como identificar letras ou números escritos à mão, cada pixel isolado não diz muita coisa

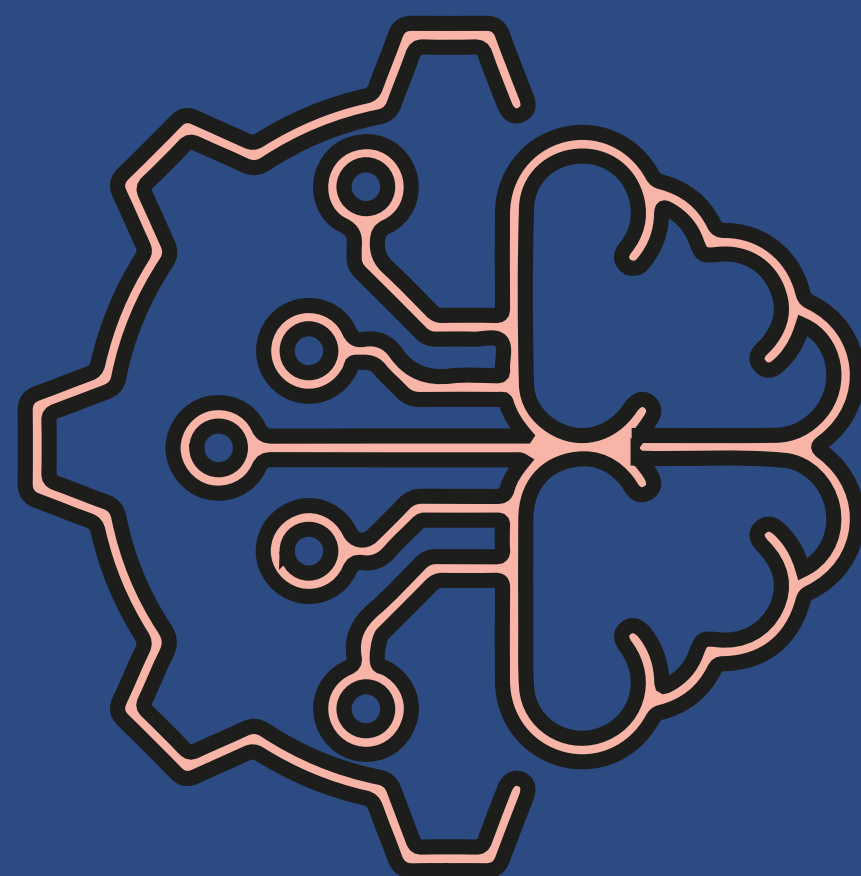
O que realmente importa são os padrões formados pela interação entre vários pixels (linhas, bordas, formas)



Não Linearidade de Parâmetros



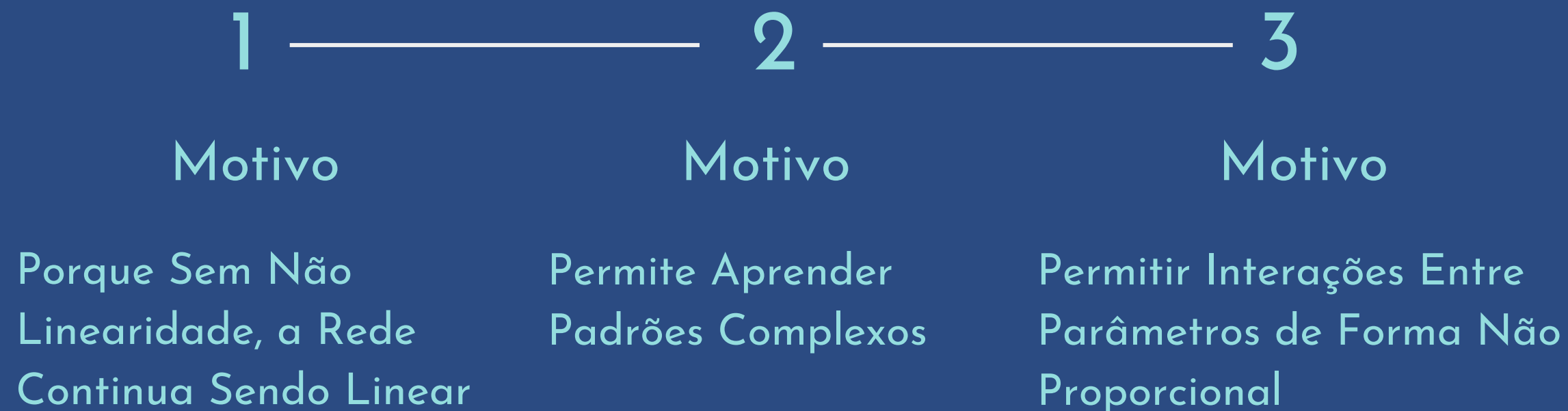
Não Linearidade dos Parâmetros



Não linearidade de parâmetros significa que o efeito de cada parâmetro no resultado final não é mais proporcional, direto e simples. Isso abre espaço para modelos muito mais ricos, capazes de capturar fenômenos reais mais complexos

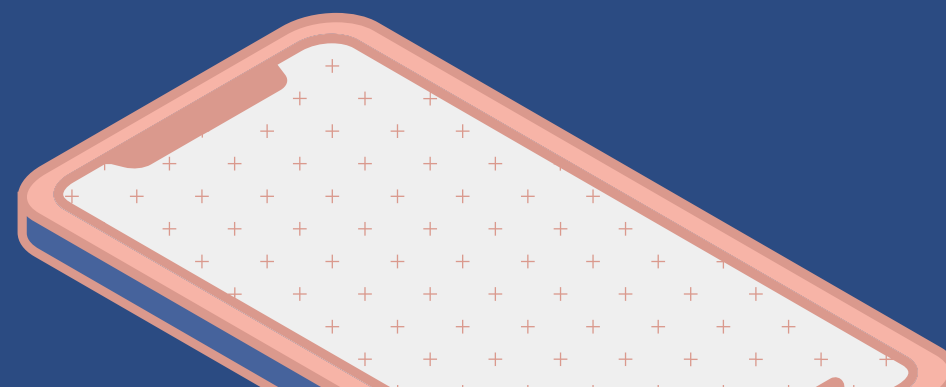
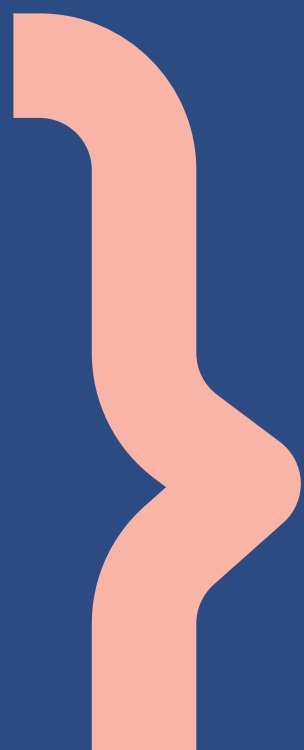
- Ela precisa envolver funções não lineares (como ReLU, Sigmoid, Tanh) que quebrem a linearidade
- Pequenas mudanças nos parâmetros podem causar grandes saltos ou oscilações no resultado

Por Que a Não Linearidade de Parâmetros é Necessária?





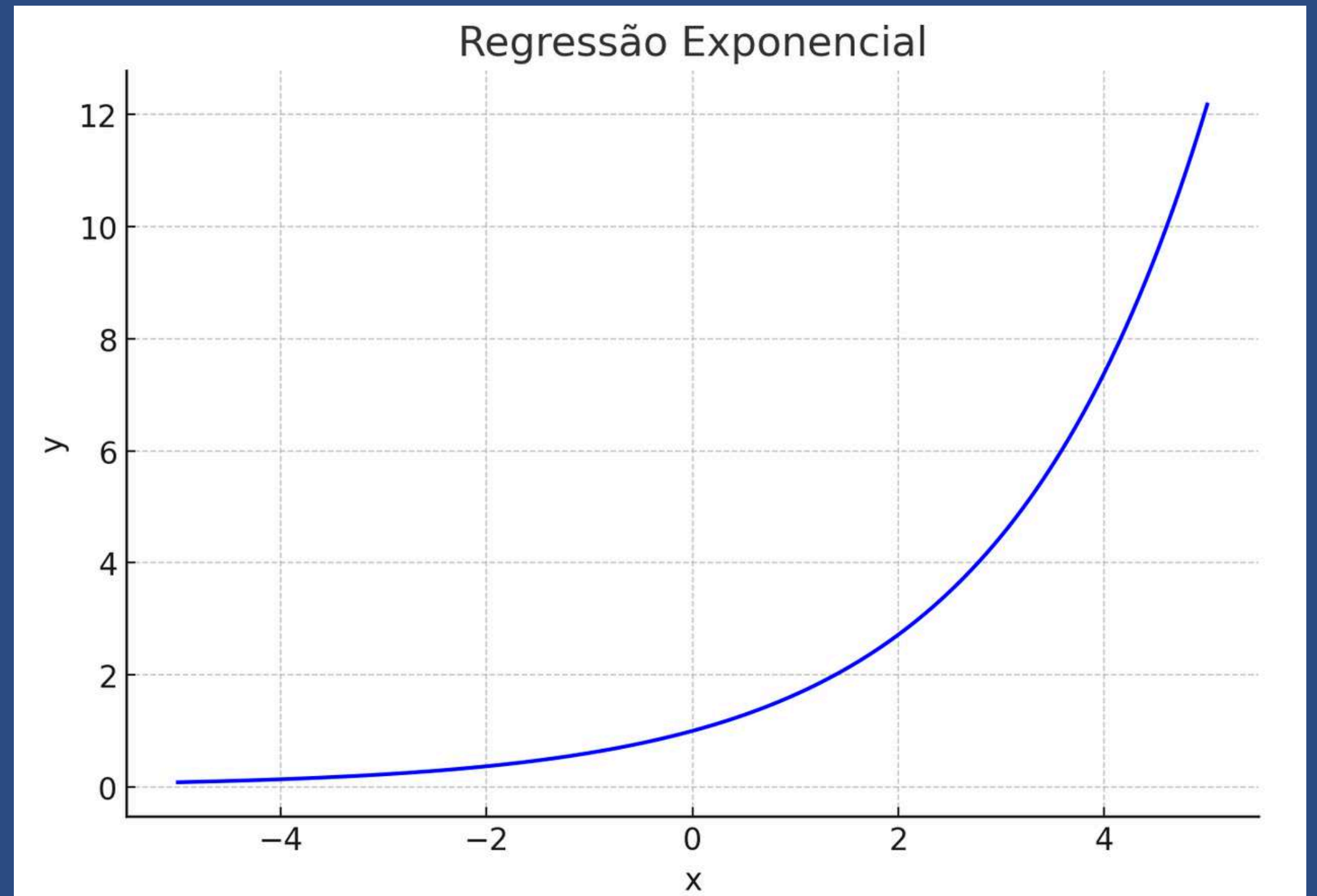
Modelos com Parâmetros Não Lineares



Regressão Exponencial

Equação: $y = \theta_0 \cdot e^{\theta_1 x}$

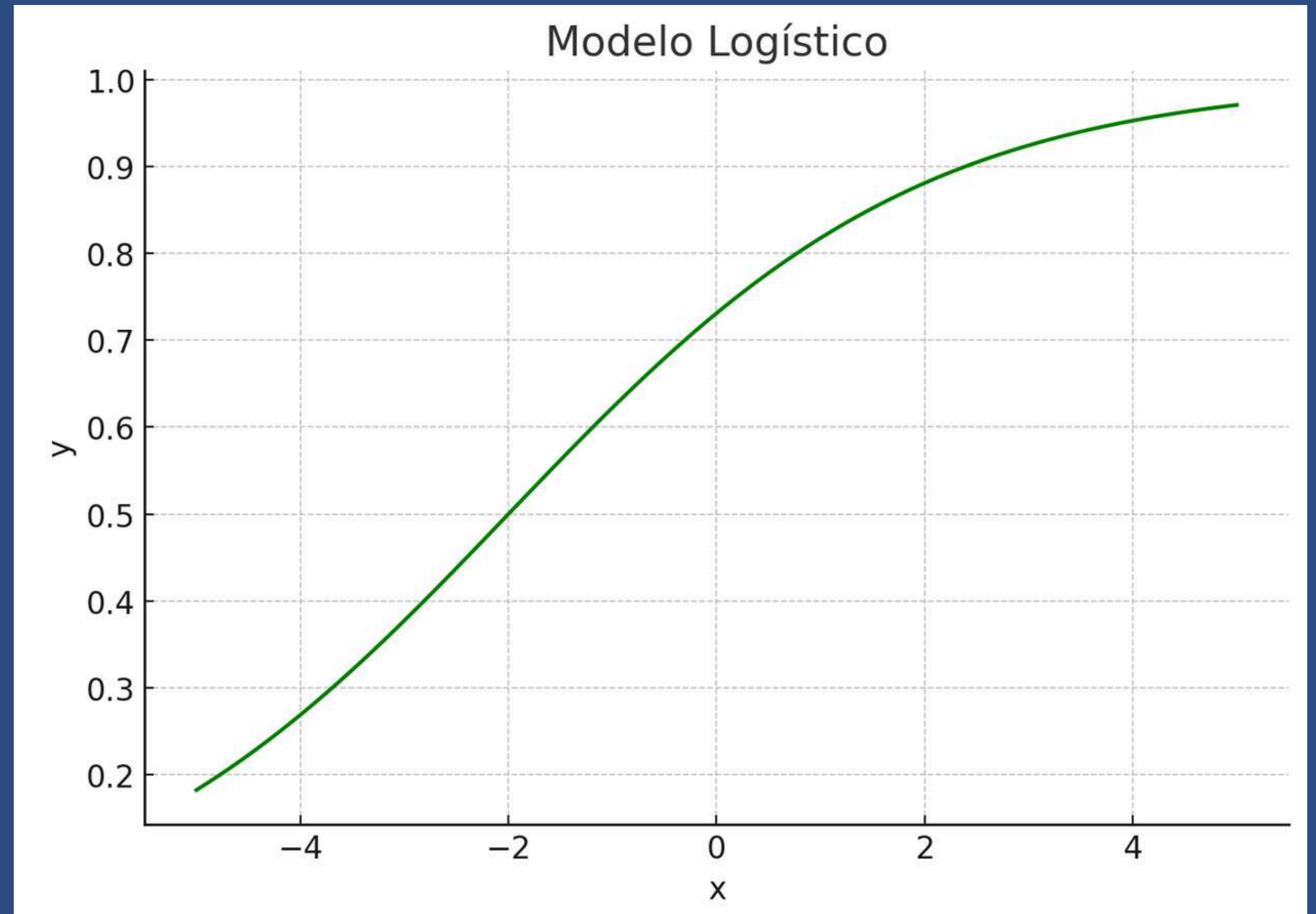
- A relação entre os parâmetros e a saída não é linear.
- Pequenas mudanças em θ_1 causam mudanças exponenciais em Y .
- Não é possível resolver com equação normal (álgebra linear); é necessário ajuste iterativo .



Modelo Logístico

Equação:
$$y = \frac{1}{1 + e^{-(\theta_0 + \theta_1 x)}}$$

- Os parâmetros θ_0 e θ_1 estão dentro de uma função sigmoide, o que quebra a linearidade.
- A saída é limitada entre 0 e 1, ideal para classificação.
- A relação entre os parâmetros e a saída não é proporcional: a sigmoide causa saturação em valores extremos.

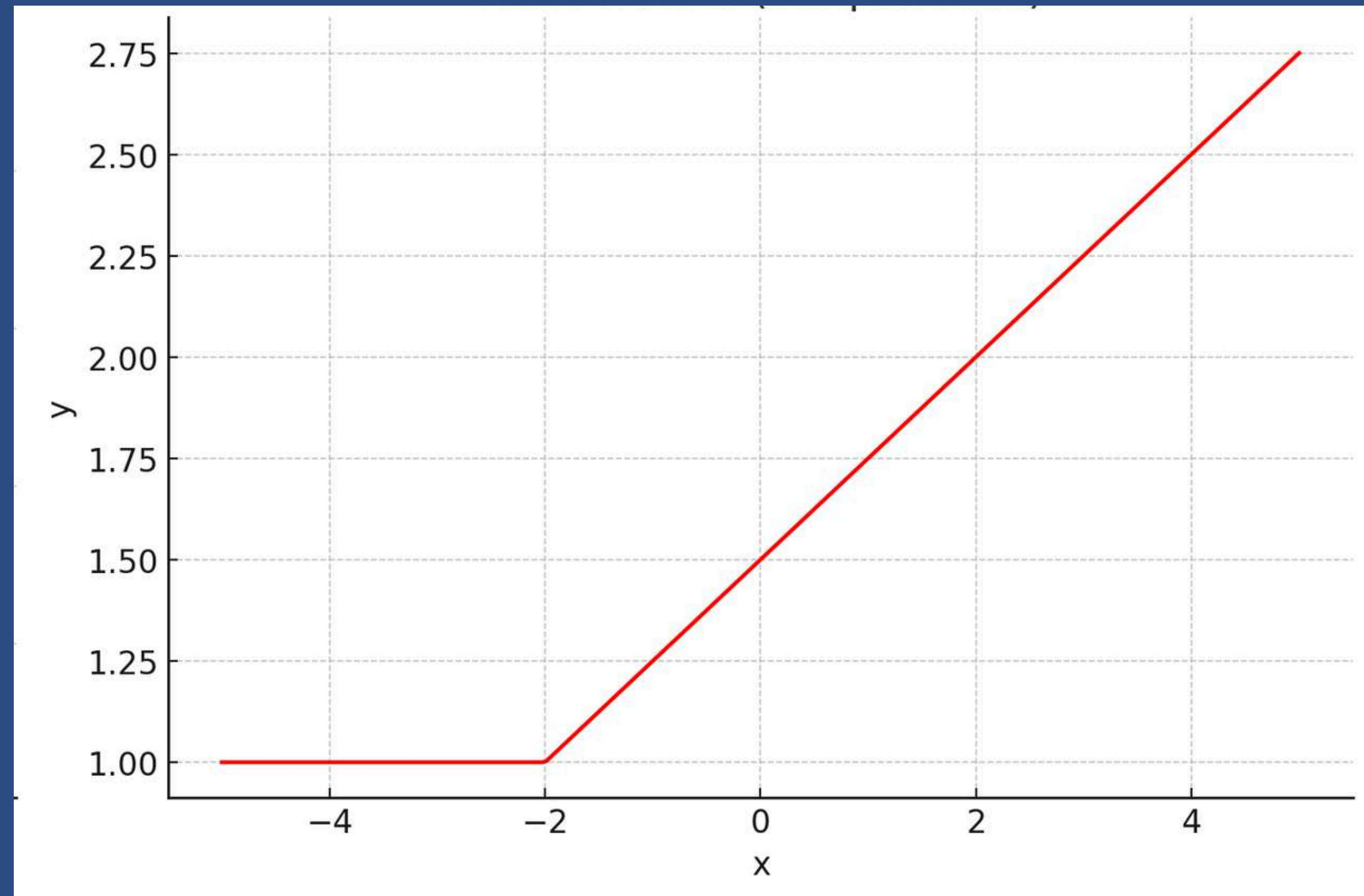


RELU

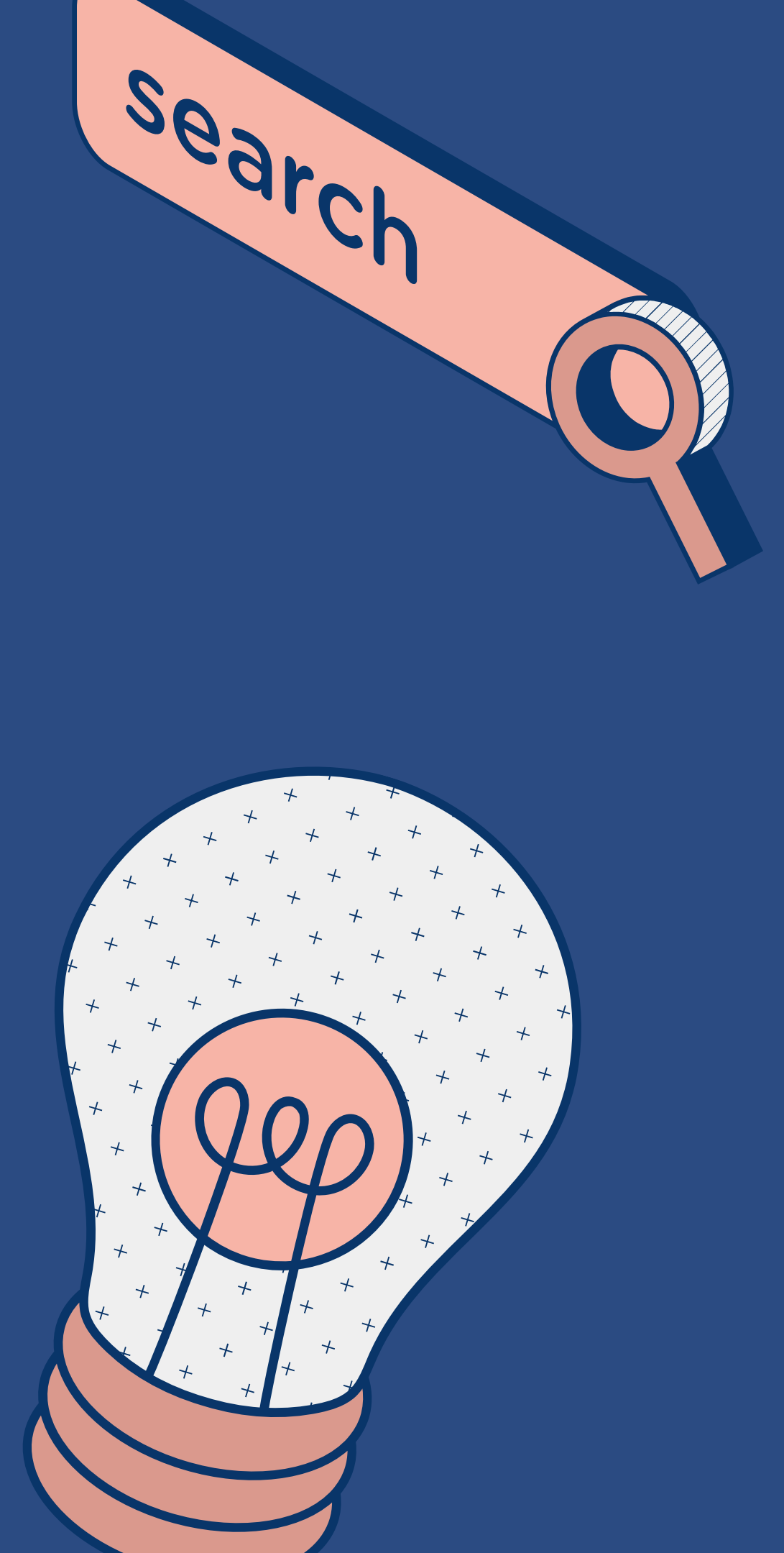
Equação:

$$y = \text{ReLU}(W_2 \cdot \text{ReLU}(W_1 x + b_1) + b_2)$$

- Cada camada aplica uma transformação linear ($Wx+b$)
- Mas depois, a função de ativação (ReLU, Sigmoid, Tanh) quebra a linearidade
- A composição de várias camadas não lineares cria um modelo extremamente poderoso



Consequências da Não Linearidade dos Parâmetros



A Função de Custo se Torna Não Convexa

Função Linear

Em uma regressão linear simples, a função de custo tem formato de tigela, onde existe apenas um mínimo global

Modelos com Parâmetros não Lineares

A função de custo não é mais uma tigela perfeita, ela pode ter múltiplos vales e picos, com múltiplos mínimos locais

Não Dá para Resolver com Fórmula Fechada

Regressão Linear

Na regressão linear, existe uma fórmula direta para calcular os coeficientes, a Equação Normal

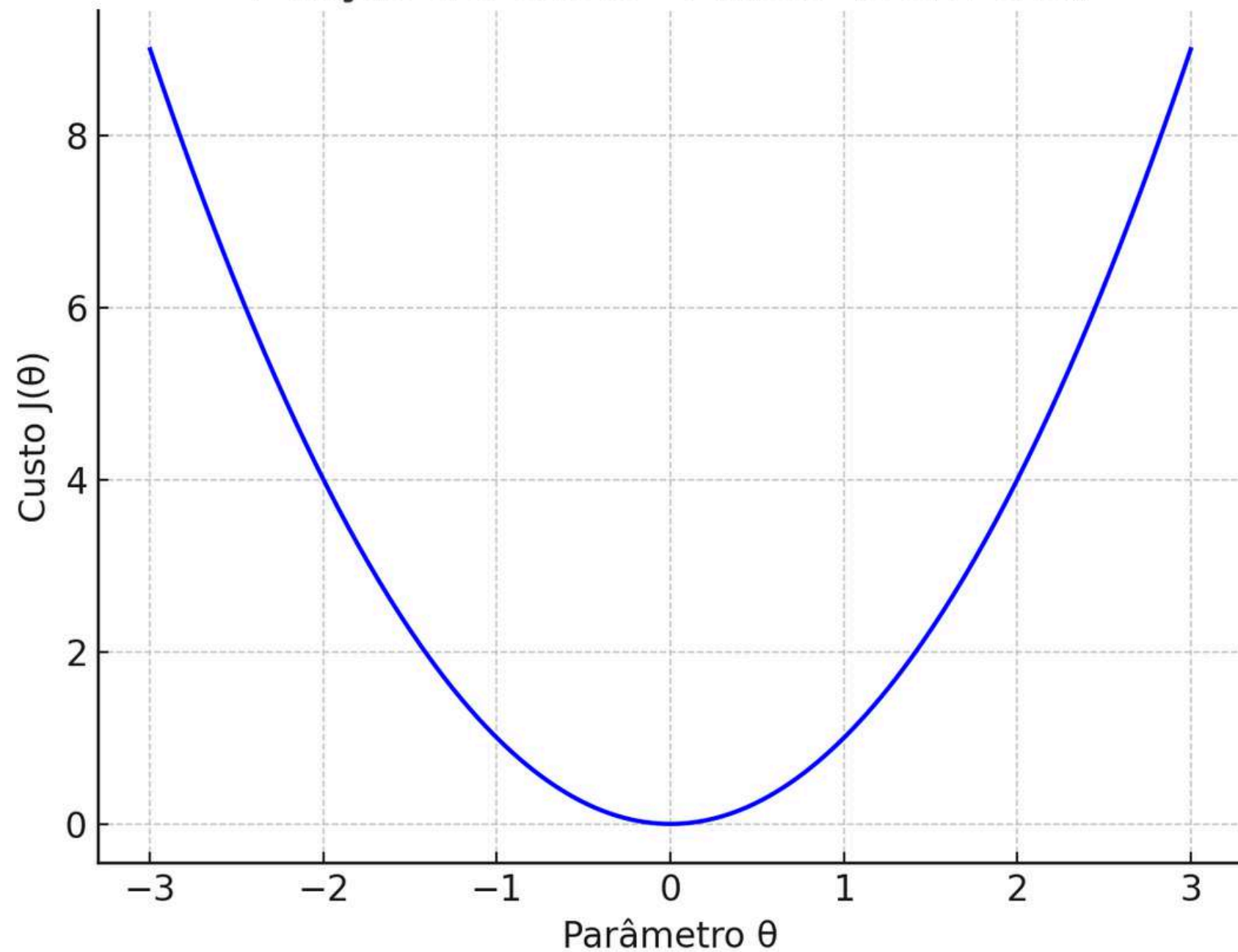
Modelos com Parâmetros não Lineares

Com não linearidade nos parâmetros, não existe essa fórmula fechada. Você precisa:

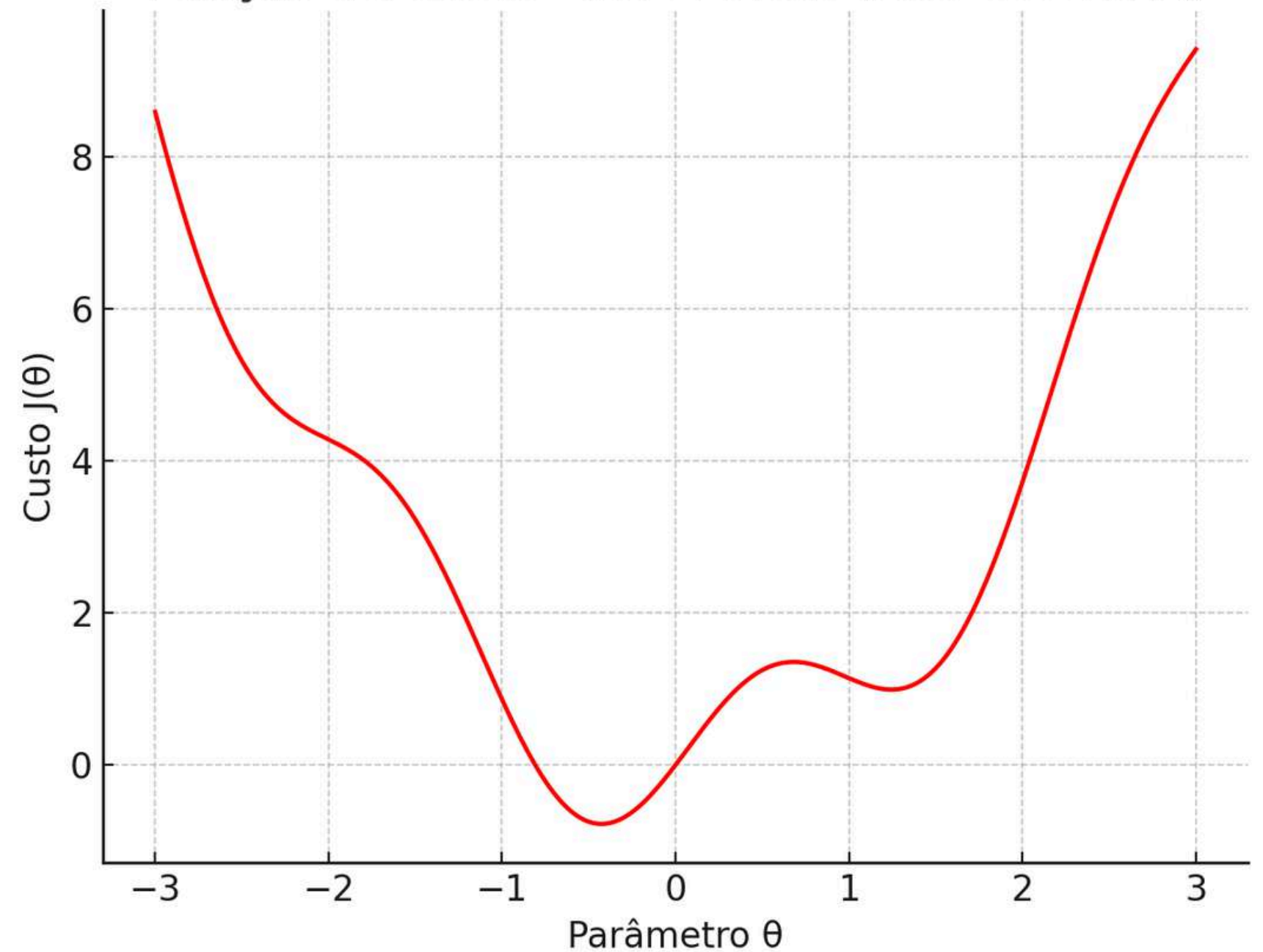
- Definir uma função de perda
- Usar métodos iterativos como gradiente descendente para ir ajustando os parâmetros aos poucos

A Função de Custo se Torna Não Convexa

Função de Custo - Linear (Convexa)



Função de Custo - Não Linear (Não Convexa)



Maior Poder de Modelagem

Flexibilidade

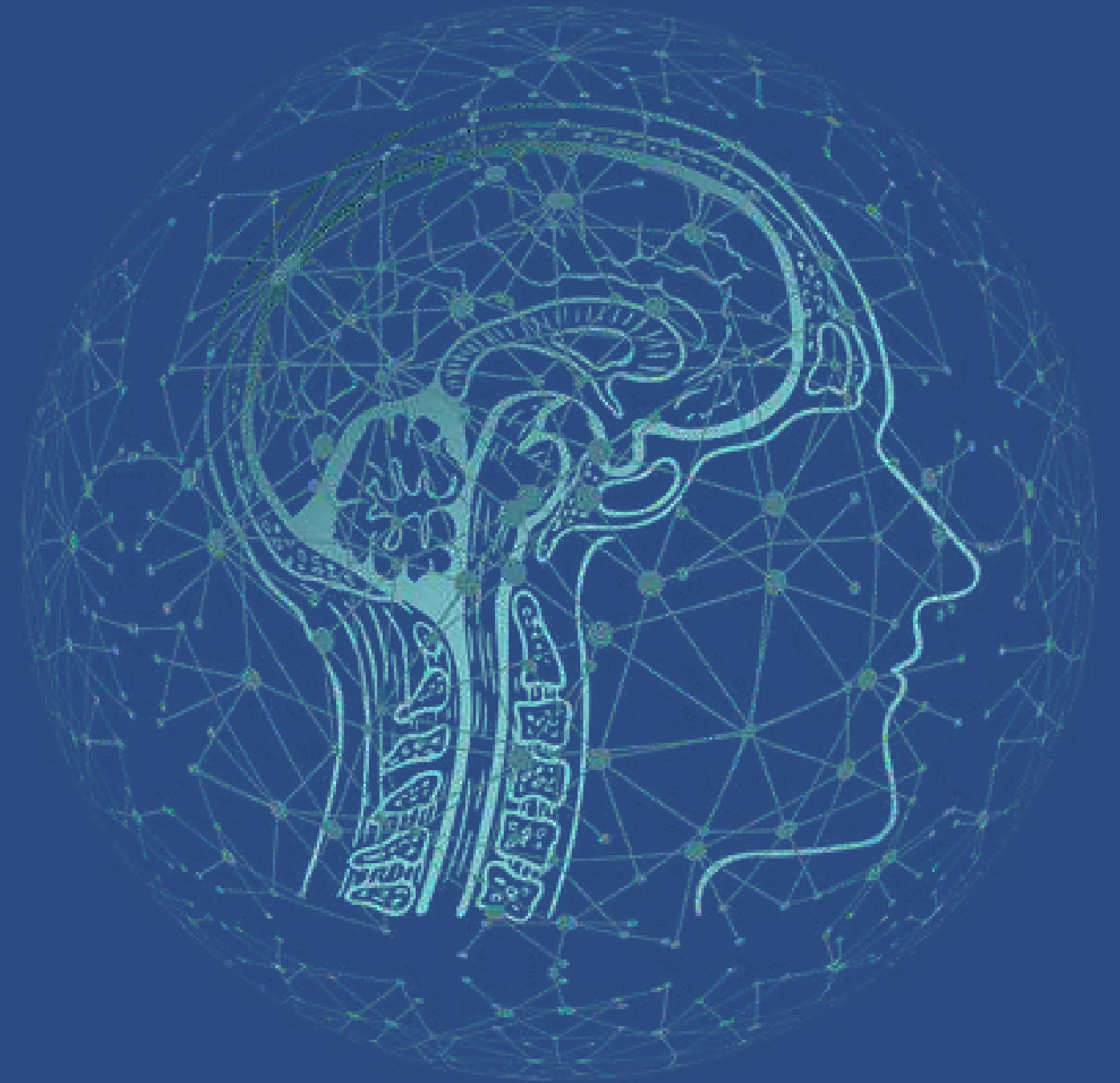
Capturam curvas complexas, saturações, efeitos exponenciais, limites superiores/inferiores

Fenômenos Possíveis de Modelar

Conseguem modelar fenômenos que modelos lineares simplesmente não conseguem, como:

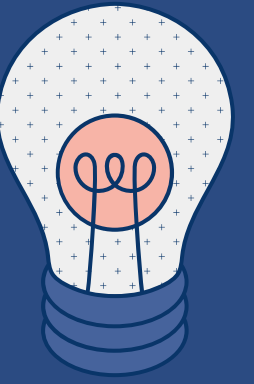
- Crescimento populacional
- Saturação de sensores
- Interações complexas em sistemas físicos ou econômicos

Redes Neurais



O que é uma Rede Neural?

- Uma rede neural é uma estrutura composta por camadas empilhadas
- Cada camada realiza duas operações básicas:
 - Transformação Linear
 - Função de Ativação



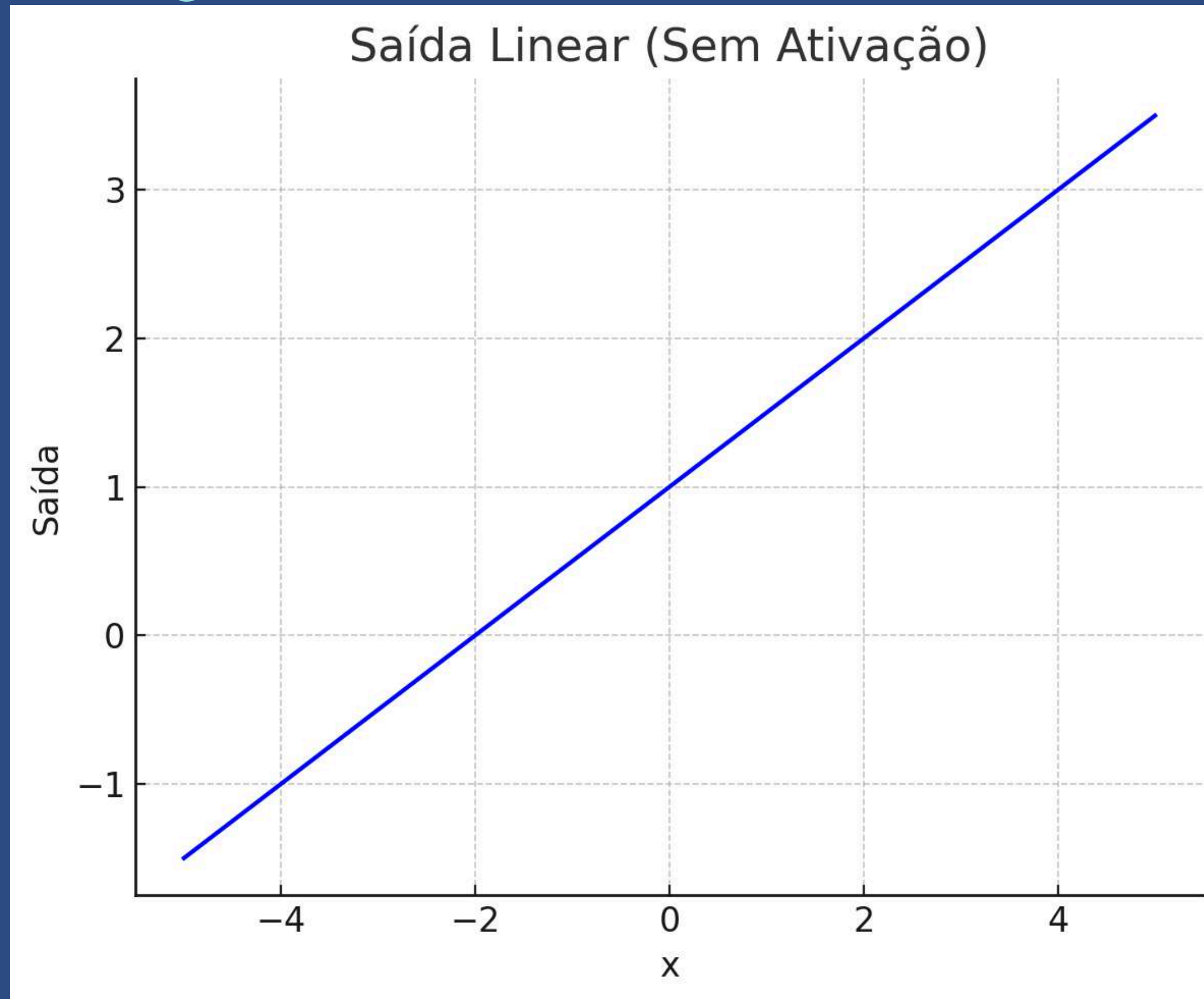
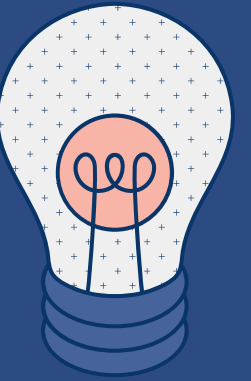
Transformação Linear

- Aqui, W (matriz de pesos) e b (vetor de bias) são os parâmetros aprendidos
- A relação entre os parâmetros e a saída Z é linear
- Se dobrarmos W , Z também dobra

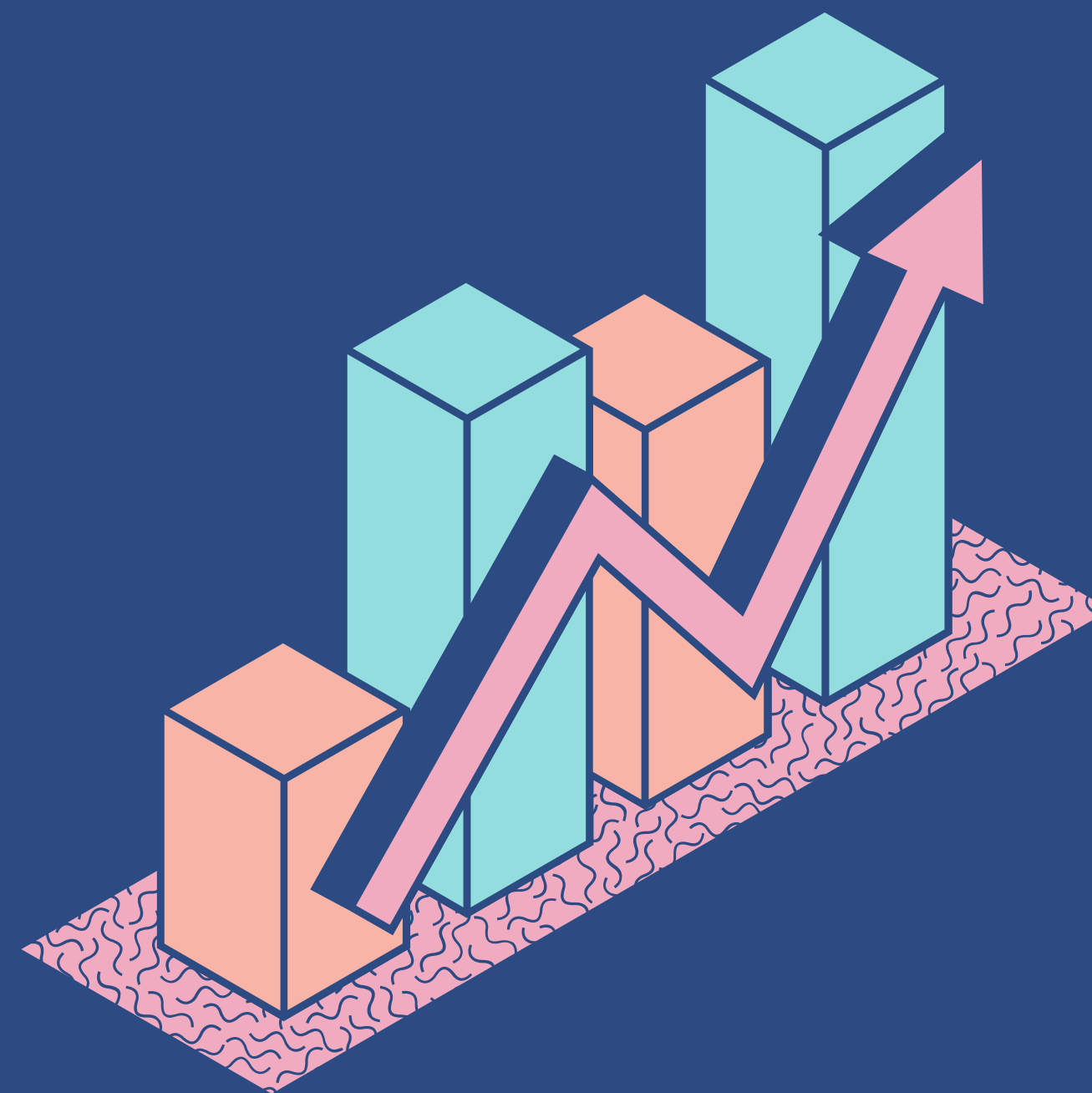
Sem nada além disso, a rede é apenas uma grande equação linear

$$z = Wx + b$$

Transformação Linear



Funções de Ativação



A Fonte da Não Linearidade no Modelo

Por que são importantes?

$$a = f(z)$$

Fonte da Não-Linearidade

Sem funções de ativação, as redes neurais não seriam neurais, seriam apenas uma grande multiplicação de matrizes

Papel da Função de Ativação

São aplicadas em cada neurônio para "quebrar" a linearidade e permitir à rede aprender padrões complexos

Permite acessar mais recursos

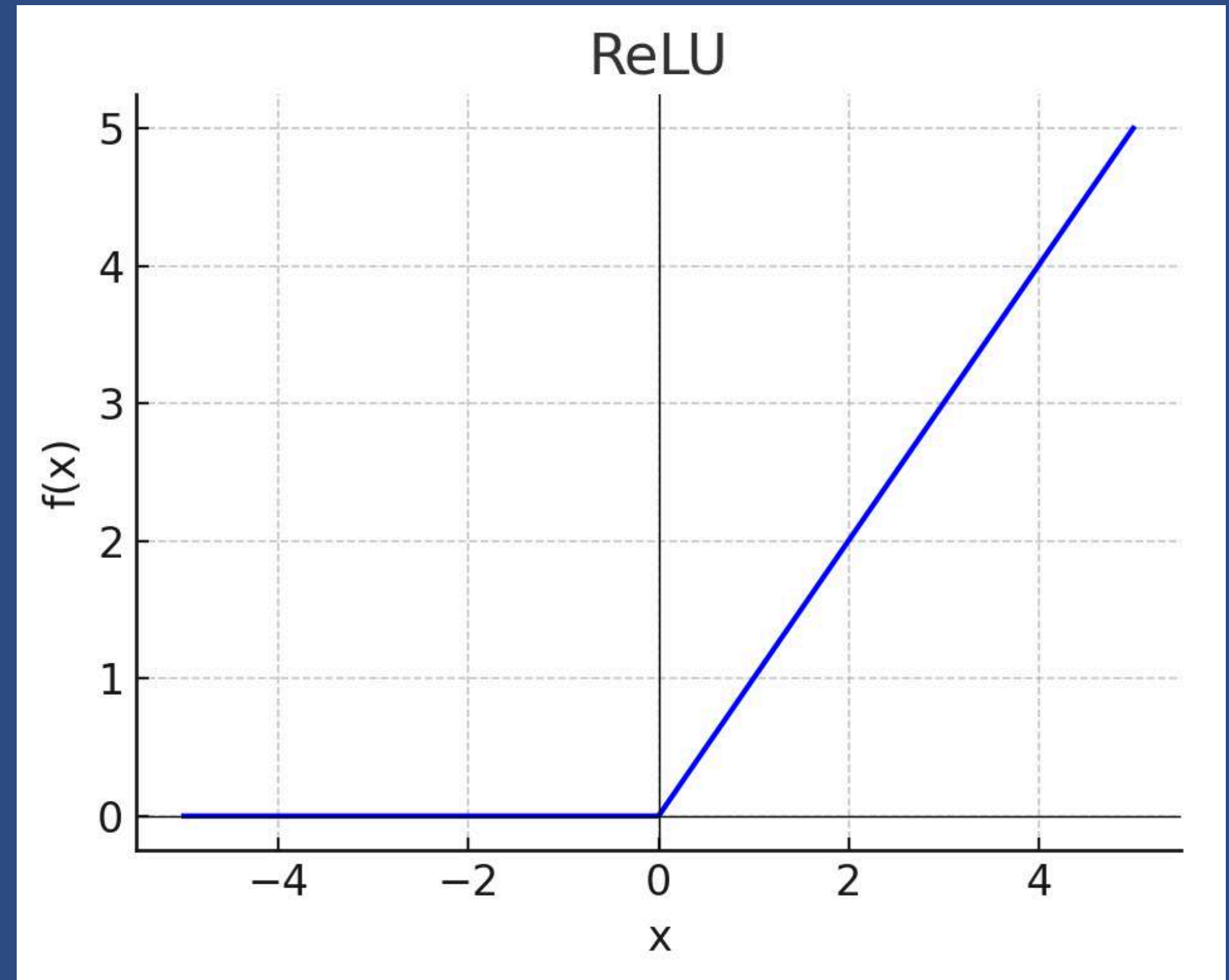
Ela permite que a rede modele relações complexas entre as variáveis e aprenda representações profundas a partir de dados brutos



Função de Ativação ReLU

Saída: $[0, \infty)$

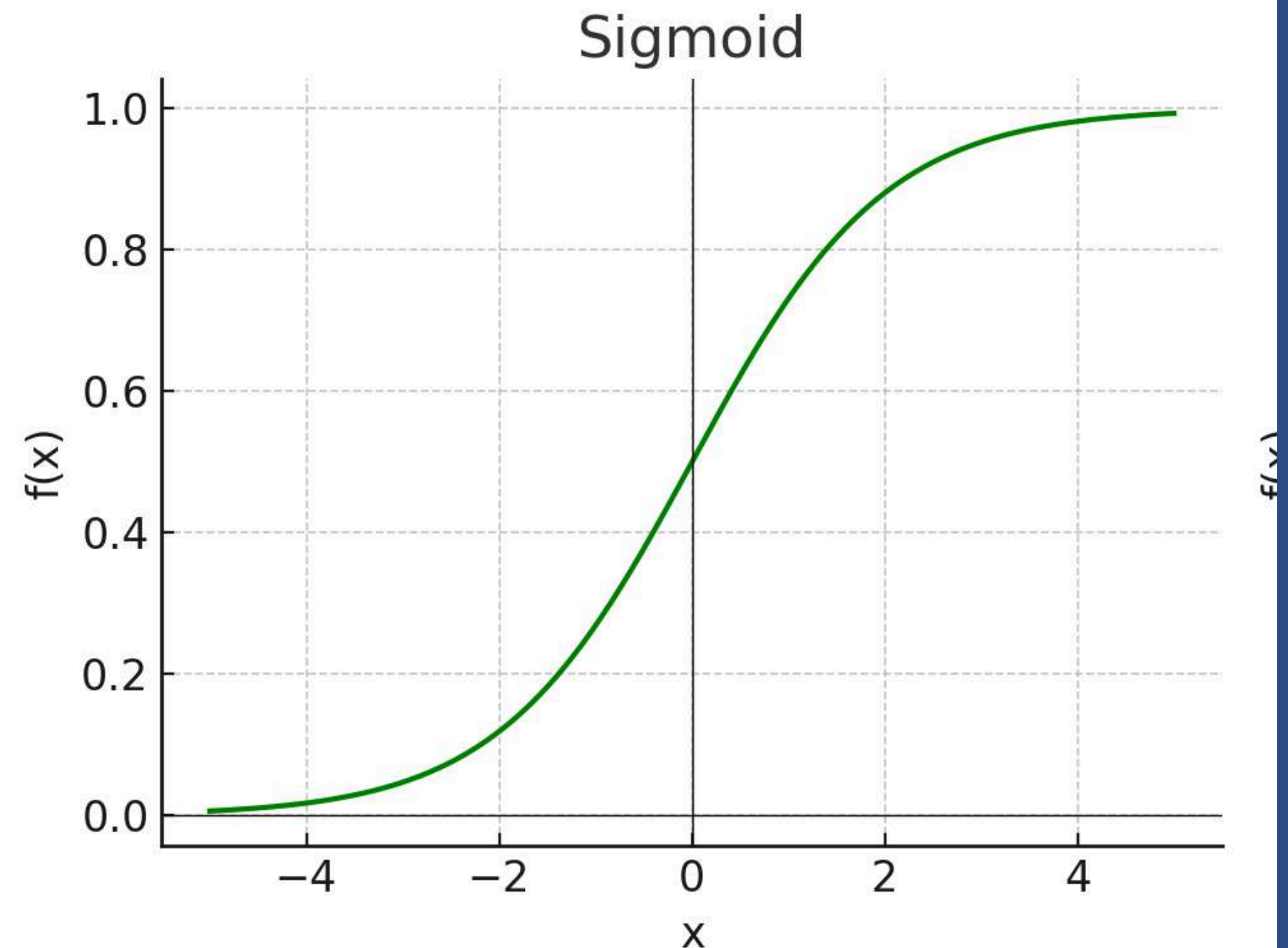
- A ReLU "dobra" a reta no zero: valores negativos são anulados, valores positivos seguem linearmente



Função de Ativação Sigmoid

Saída: $[0,1]$

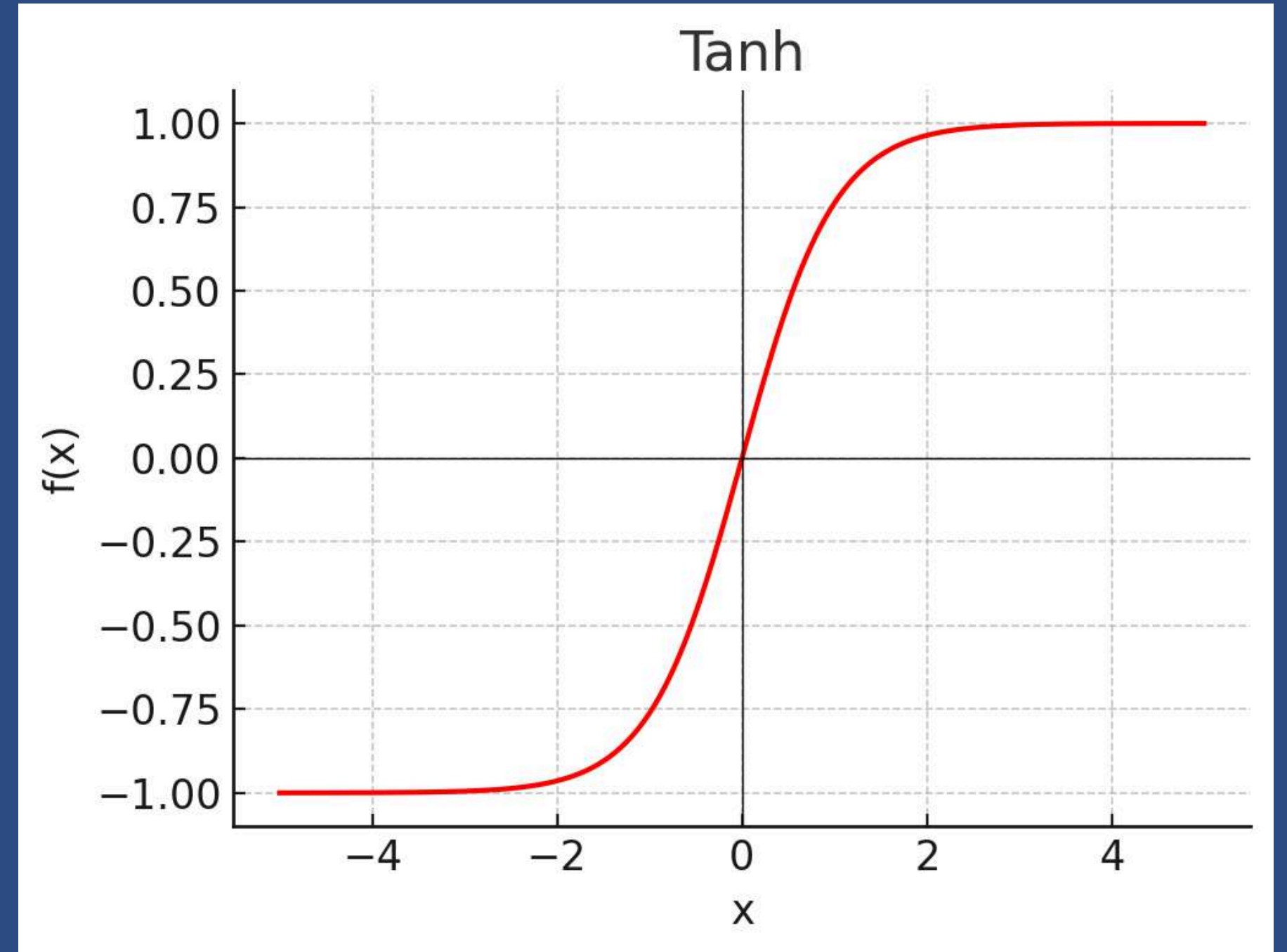
- Uma curva em "S", logística
- Comprime a saída entre 0 e 1



Função de Ativação Tanh

Saída : $[-1,1]$

Curva em S centrada em zero



Teorema da Aproximação Universal



Definição do Teorema

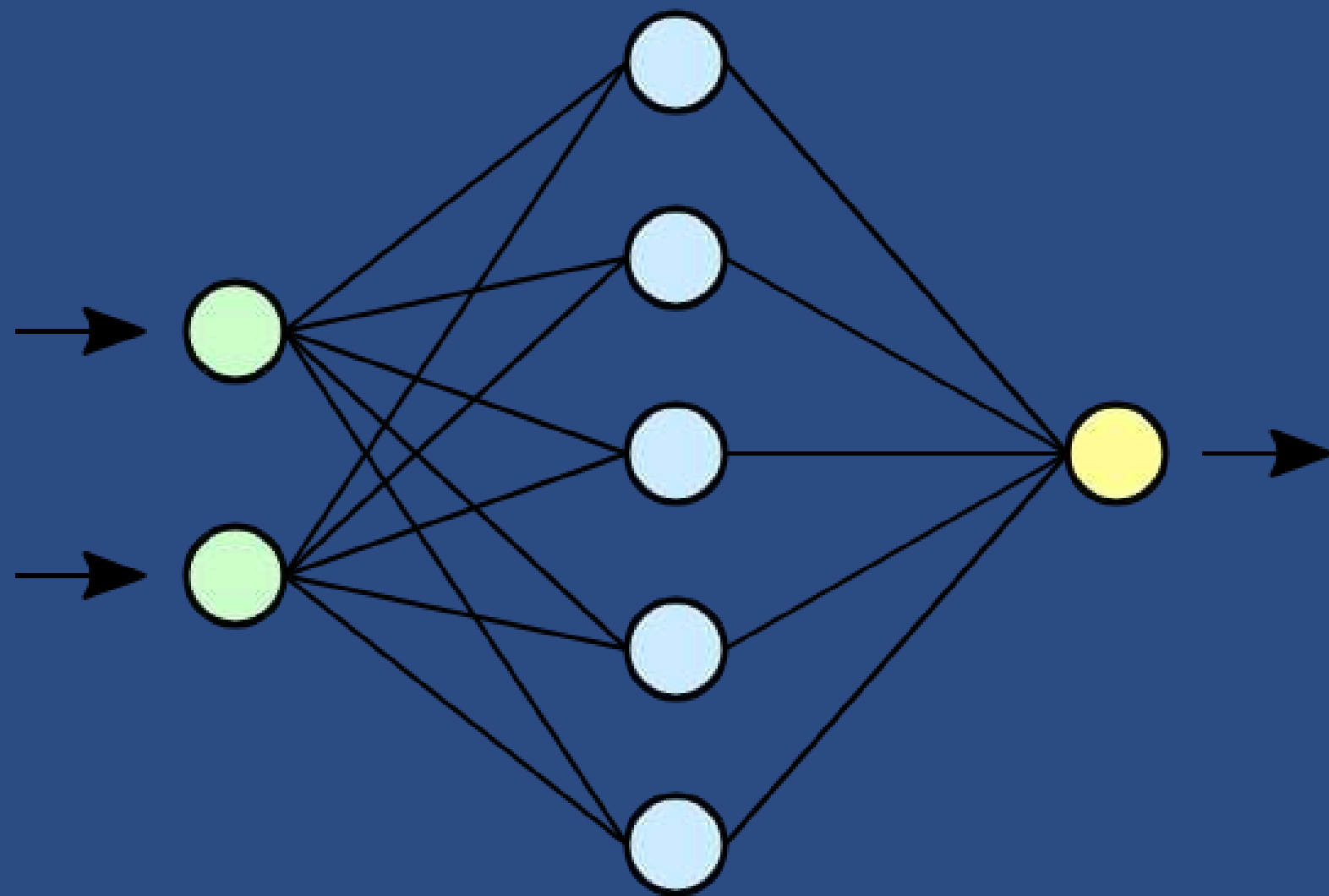


Uma rede neural com pelo menos uma camada oculta e funções de ativação não linear pode aproximar qualquer função contínua definida em um intervalo compacto, com qualquer nível de precisão desejada

Redes neurais têm a capacidade teórica de aprender qualquer padrão, desde que haja neurônios suficientes

A Rede Neural só tem essa capacidade só se ela tiver funções de ativação não lineares

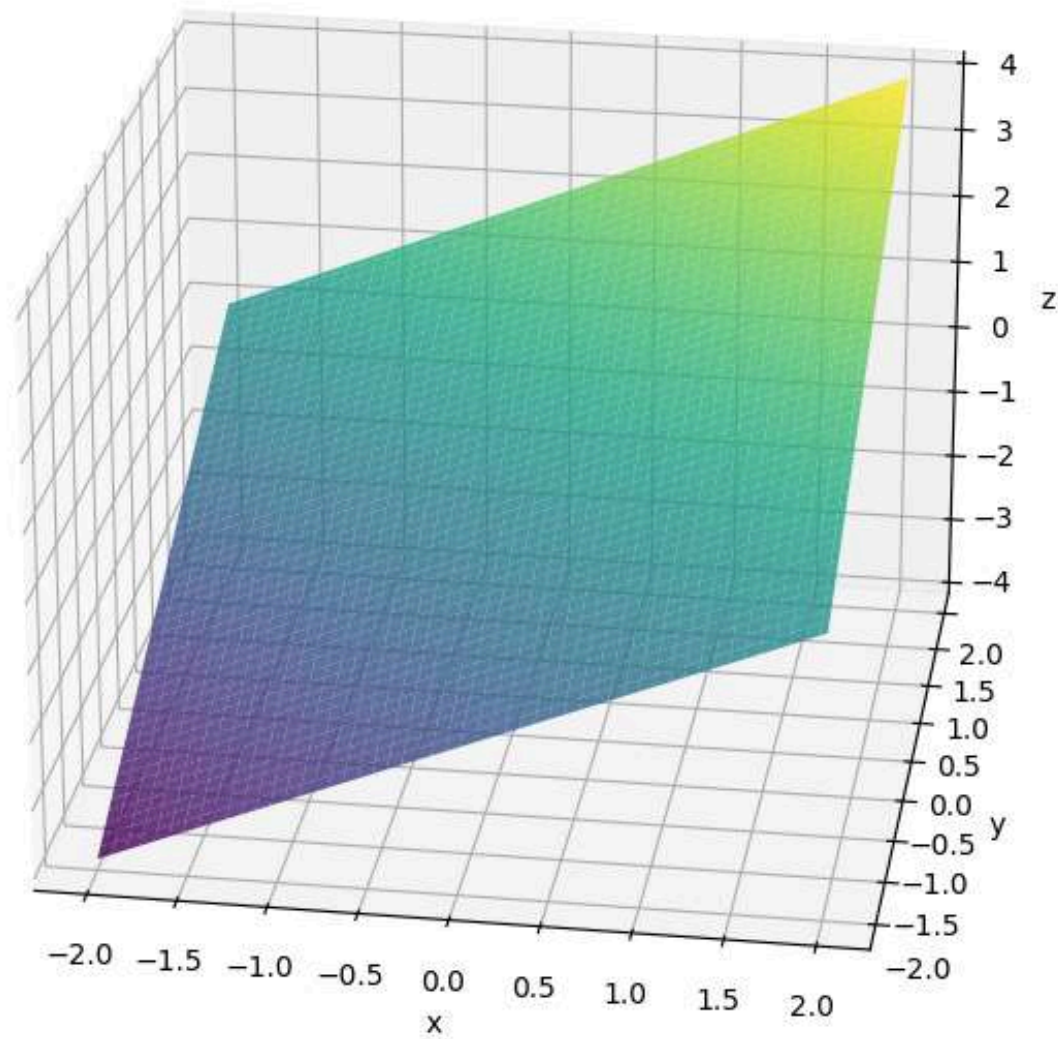
Importância do Teorema



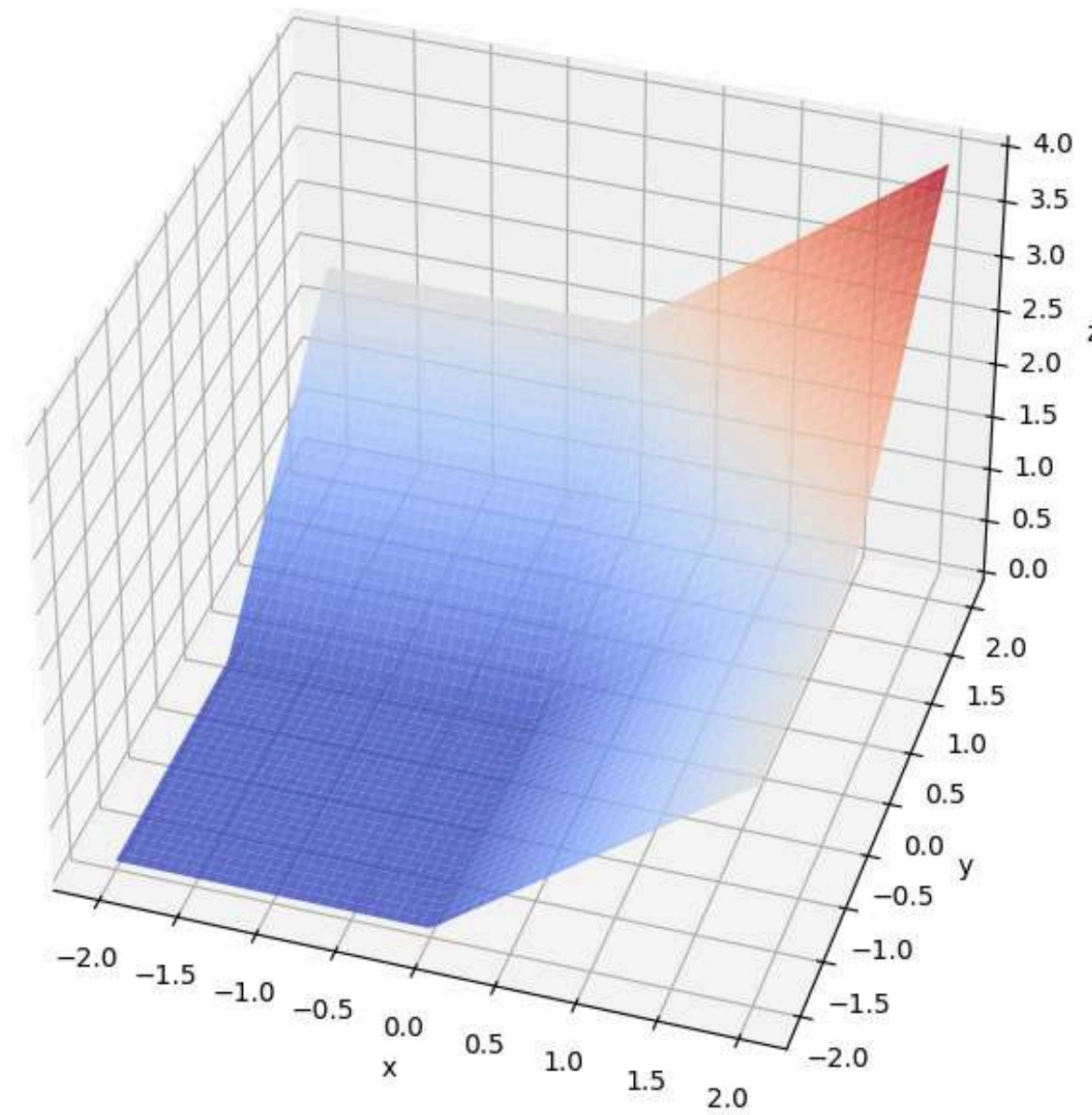
Garante o Potencial das Redes Neurais

- A base teórica que justifica o uso de redes neurais para resolver problemas complexos e variados
- Traz Segurança Teórica para a Modelagem
- O desafio não está mais na capacidade da rede, mas em como treiná-la e configurá-la bem

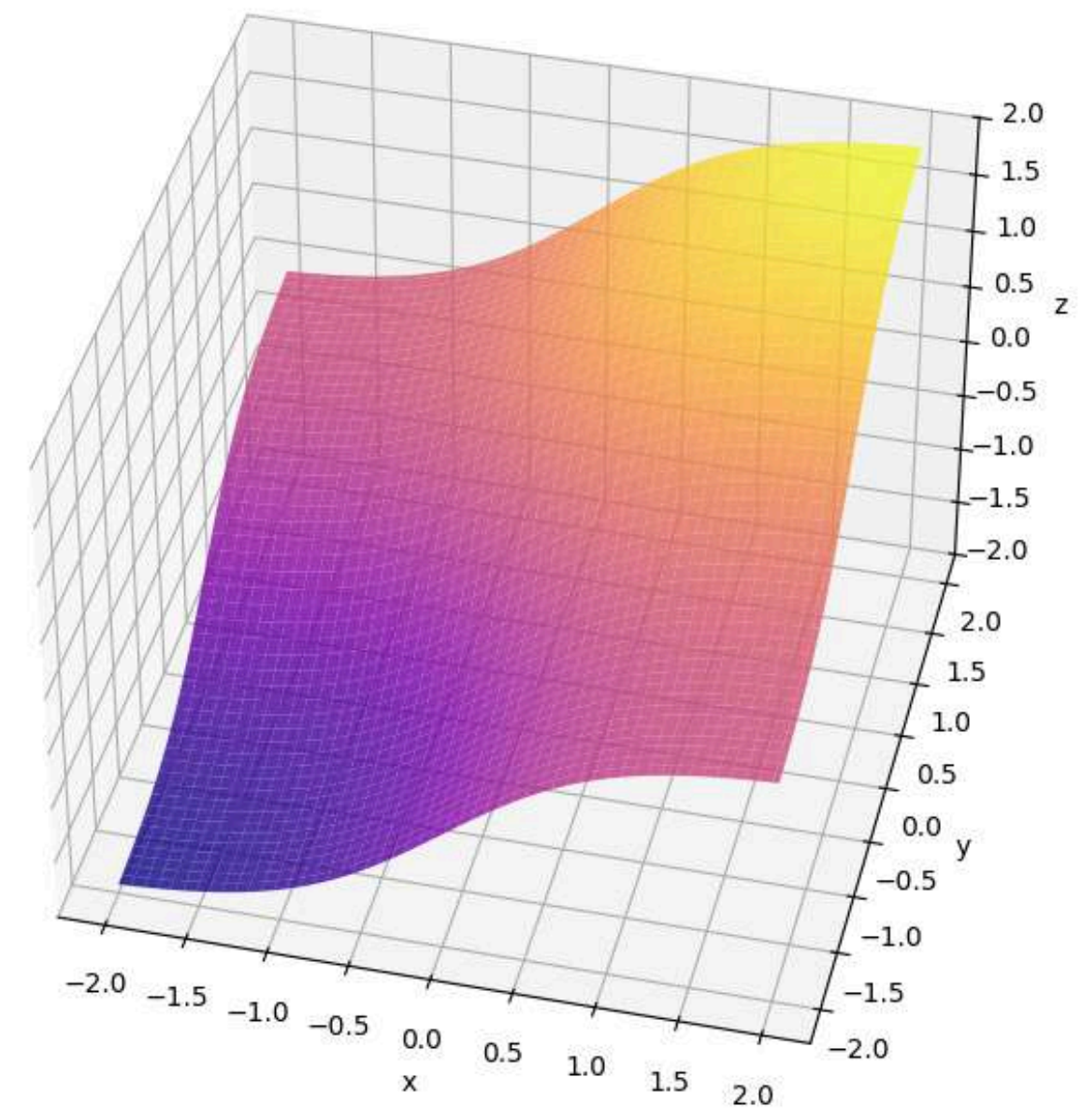
Sem Ativação (Linear - Blocos Retos)



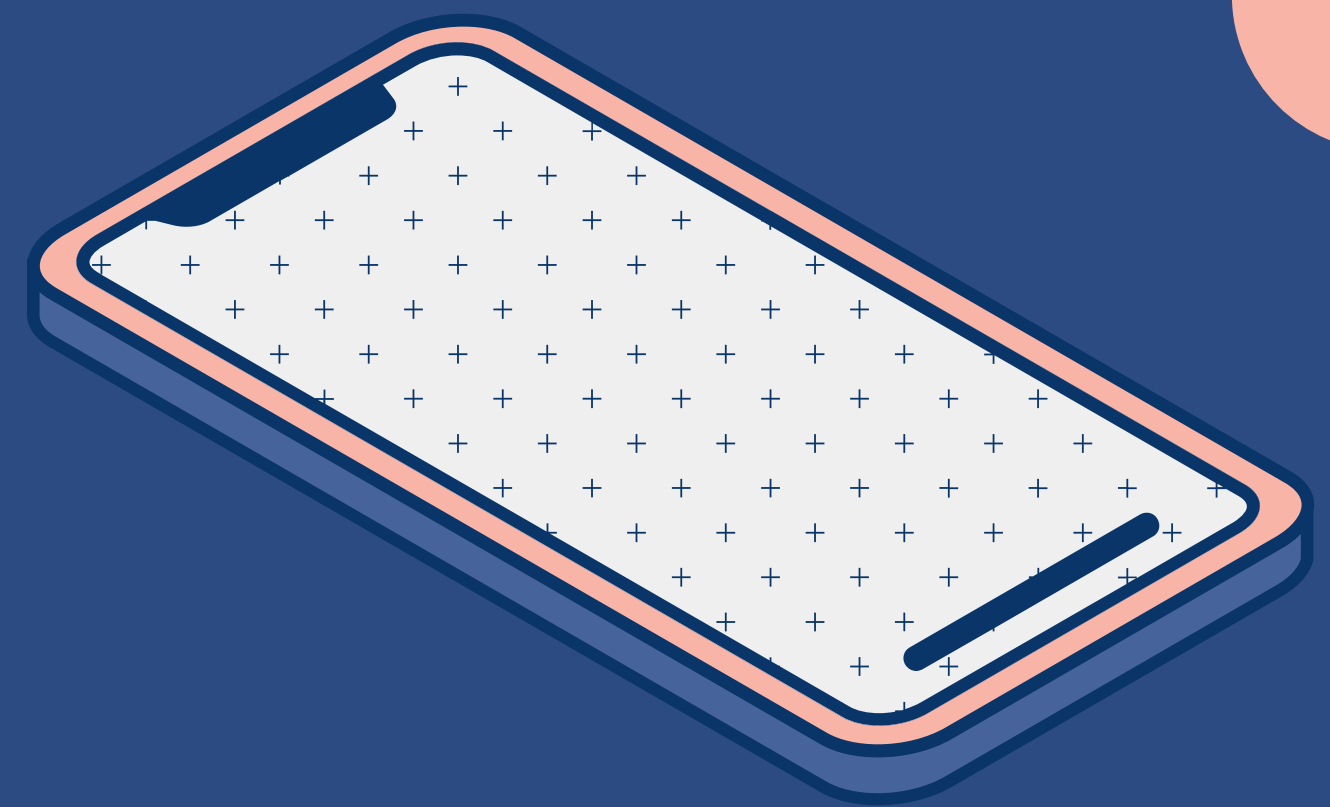
Com ReLU (Dobras)



Com Tanh (Curvas Suaves)



Implicações e Limitações



Implicações para a Otimização

A não linearidade nos parâmetros dificulta a otimização



A função de custo pode ter múltiplos mínimos locais e máximos

Problemas como saddle points e planos rasos (plateaus) aparecem, complicando a convergência

Exige técnicas de ajuste de aprendizado, como:

- Inicialização cuidadosa dos pesos (Xavier, He)
- Métodos avançados de otimização: momentum, Adam, RMSProp

Overfitting e Generalização

➤ Overfitting

Quando o modelo fica muito flexível, ele pode aprender demais:

- Não só os padrões reais do dado, mas também os ruídos e variações aleatórias
- O modelo vai muito bem no treino, mas vai mal em novos dados

➤ Regularização

Para evitar overfitting, usamos regularização:

- L2
- Dropout

➤ Desafio

Queremos um modelo flexível o suficiente para aprender o padrão, mas não tão flexível a ponto de aprender o ruído

○ Resto da apresentação ocorrerá
no Ipython!

