Inteligência Artificial

Aula 15 – Redes Neurais Artificiais

Ciência da Computação
Prof. Maiko Spiess

Roteiro

- Antecedentes
- Perceptron

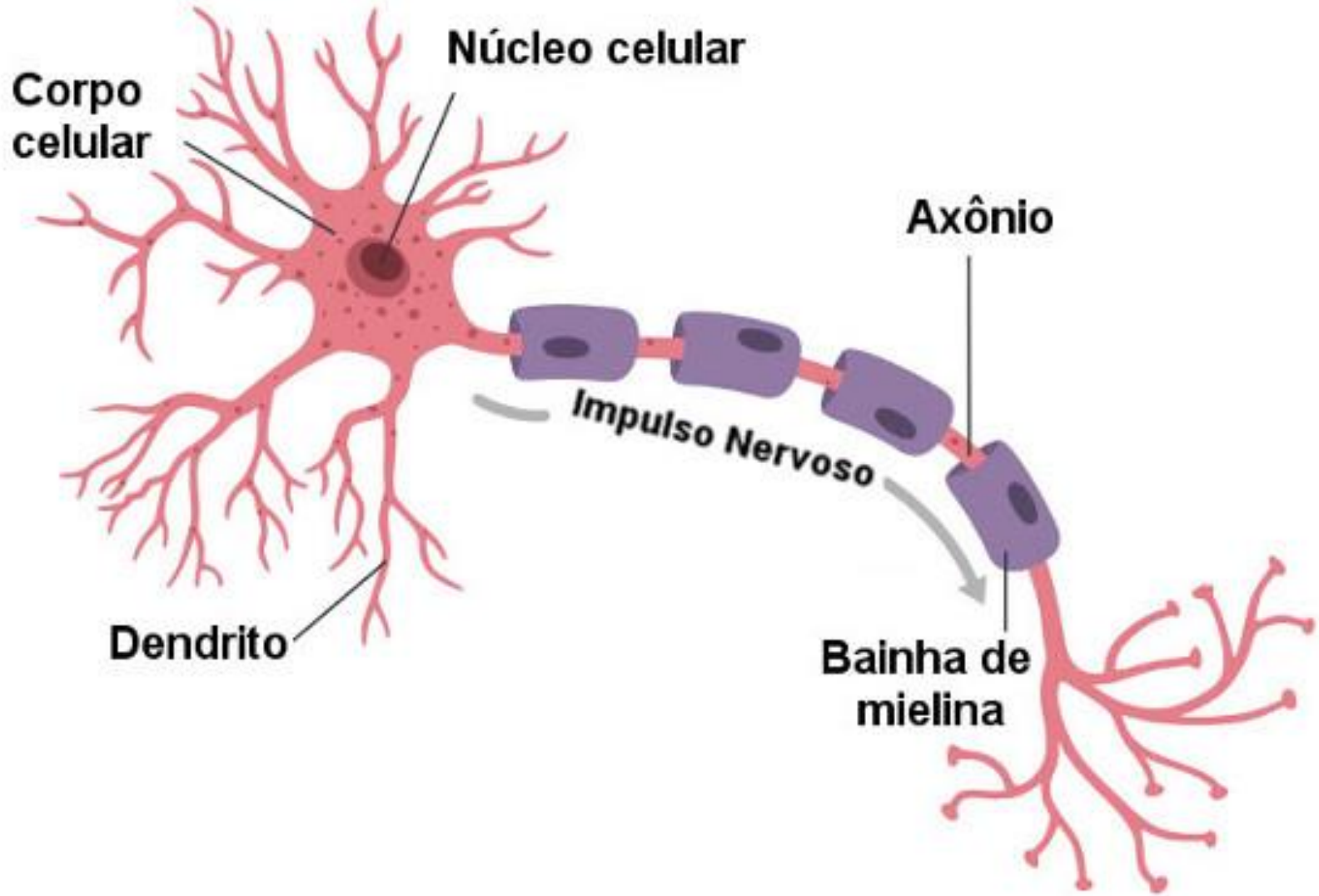
Antecedentes

O neurônio biológico

A inspiração conceitual para as redes neurais artificiais é o sistema nervoso biológico.

A unidade básica deste sistema é o neurônio, uma célula altamente especializada que possui três funções primárias:

1. Receber sinais (informação)
2. Integrar os sinais de entrada (para determinar se a informação deve ser repassada)
3. Comunicar sinais às células alvo (outros neurônios, músculos ou glândulas).


Corpo celular
Núcleo celular
Axônio
Impulso Nervoso
Dendrito
Bainha de mielina

Componentes do neurônio biológico

- **Dendritos:** extensões ramificadas que atuam como os terminais de entrada da célula, recebendo sinais de outros neurônios.
- **Soma (Corpo Celular):** o centro metabólico e de processamento da célula. Ele coleta e integra os sinais elétricos recebidos pelos dendritos.
- **Axônio:** uma longa projeção única que funciona como o canal de saída da célula, transmitindo o sinal processado para longe do soma.
- **Sinapse:** a junção funcional e especializada entre o terminal do axônio de um neurônio (célula pré-sináptica) e o dendrito de outro (célula pós-sináptica).

Na maioria das sinapses, a informação é transmitida quimicamente através de mensageiros chamados neurotransmissores.

Funcionamento do neurônio biológico

O processo de disparo neuronal é fundamental para a computação biológica. Os neurotransmissores liberados na fenda sináptica podem ser **excitatórios** (aumentando a probabilidade do neurônio pós-sináptico disparar) ou **inibitórios** (diminuindo-a).

O neurônio pós-sináptico, em seu núcleo, integra espacial e temporalmente todos esses sinais de entrada. **limiar de disparo (*threshold*)**Se a excitação líquida combinada ultrapassar um, o neurônio dispara um "potencial de ação"—um impulso elétrico de magnitude fixa que viaja pelo axônio. Este é um **evento "tudo ou nada" (*all-or-none*)**.

A analogia computacional

É crucial entender que o objetivo de uma Rede Neural Artificial **não é** simular realisticamente a biologia. O neurônio artificial é uma abstração computacional da função biológica.

Toda a complexidade biológica, como o papel das células da glia (astrócitos, microglia), que nutrem e dão suporte aos neurônios, a química específica dos neurotransmissores ou o isolamento elétrico fornecido pela bainha de mielina, é intencionalmente omitida.

A analogia computacional

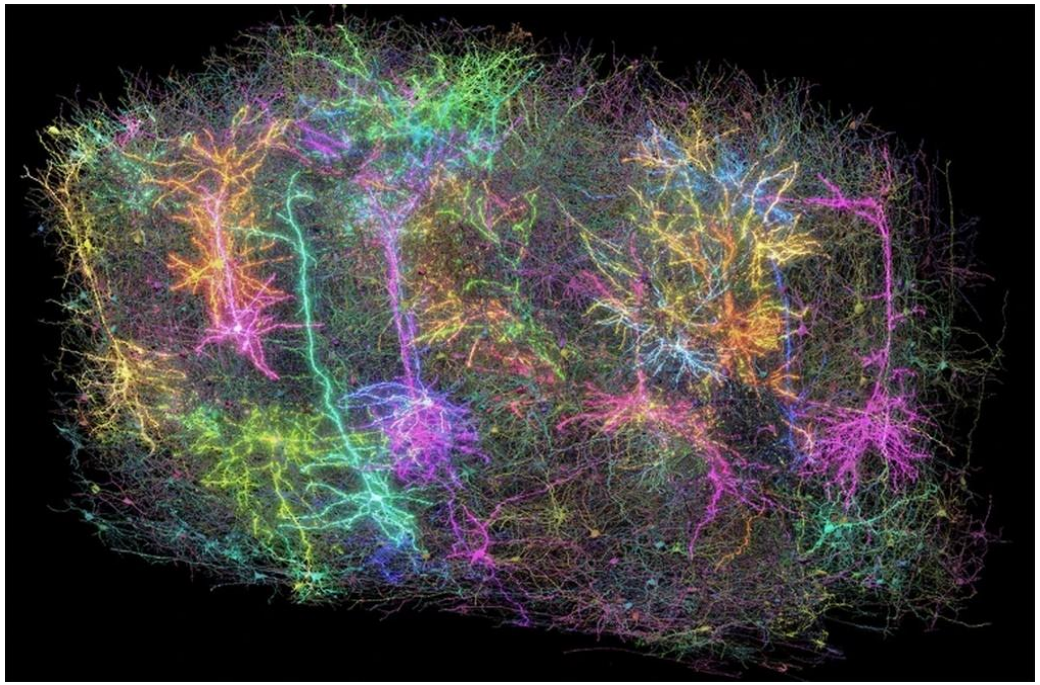
O que os pioneiros da computação extraíram foi o **princípio funcional mínimo: sinais de entrada são ponderados, somados e, em seguida, passam por um limiar de decisão.**

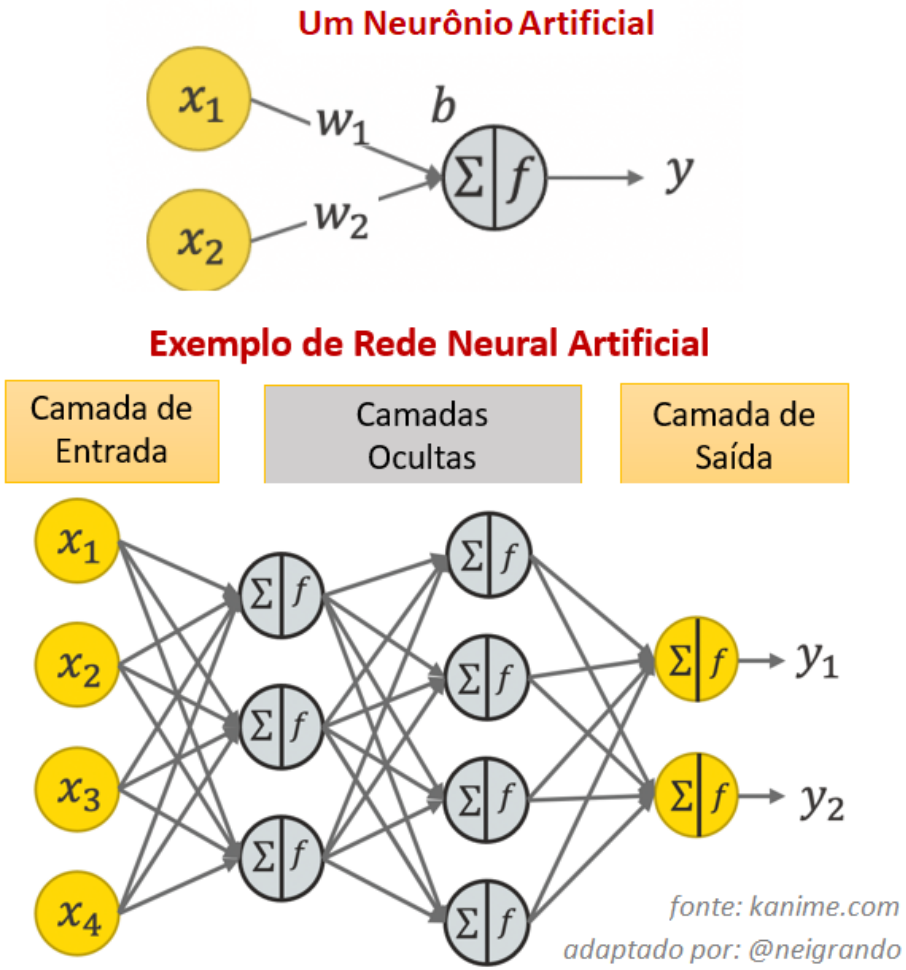
Nesta analogia, o conceito biológico de sinapses "excitatórias" e "inibitórias" é capturado pelo sinal matemático do peso.

Um **peso positivo** ($w_i > 0$) representa uma sinapse excitatória, enquanto um **peso negativo** ($w_i < 0$) representa uma sinapse inibitória.

Toda a complexidade da neuroquímica é, portanto, reduzida a uma aritmética simples, tornando o "aprendizado" (a mudança nesses pesos) um problema matematicamente tratável.

Conceitos Principais





Primeiros desenvolvimentos

O Neurônio como dispositivo lógico: McCulloch e Pitts (1943)

O primeiro grande marco formal foi o artigo de 1943, "A Logical Calculus of the Ideas Immanent in Nervous Activity", do neurofisiologista Warren McCulloch e do lógico Walter Pitts.

Eles propuseram um modelo de neurônio artificial baseado em cinco pressupostos simplificadores:

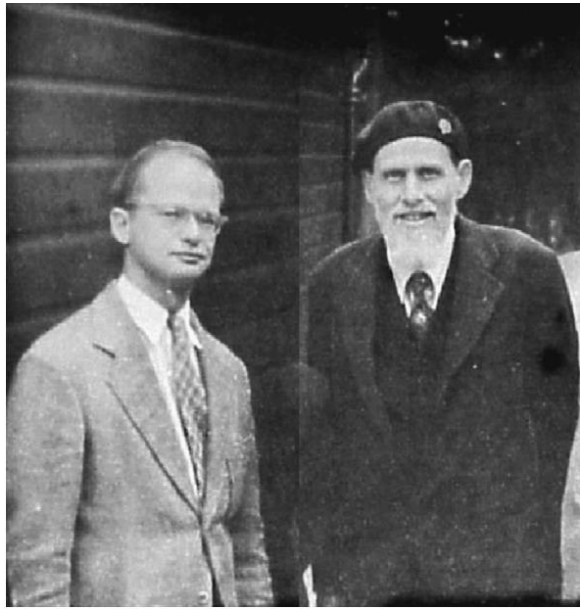
Primeiros desenvolvimentos

- A atividade do neurônio é um processo "tudo ou nada" (all-or-none).
- Um número fixo de sinapses (um limiar) deve ser excitado dentro de um período de tempo para que o neurônio dispare.
- O único atraso significativo no sistema é o atraso sináptico.
- A atividade de qualquer sinapse inibitória impede absolutamente o disparo do neurônio naquele momento.
- A estrutura da rede (as conexões e seus limiares) não muda com o tempo.

Primeiros desenvolvimentos

A grande contribuição do modelo McCulloch-Pitts (M-P) foi a prova de que essa unidade simples, quando conectada em redes, poderia funcionar como portas lógicas (AND, OR, NOT).

Esse trabalho foi imensamente influente, pois conectou pela primeira vez a neurofisiologia à Teoria da Computação de Turing e influenciou diretamente pioneiros como John von Neumann.



A Regra de Hebb (1949)

O modelo M-P era poderoso em sua capacidade computacional, mas era estático. Seu quinto pressuposto (estrutura fixa) era uma limitação óbvia para modelar um sistema biológico que aprende.

Seis anos depois, em 1949, o psicólogo Donald O. Hebb, em seu livro "The Organization of Behavior", propôs o primeiro mecanismo de aprendizado plausível. Hebb não propôs um modelo matemático formal, mas sim um postulado biológico, agora conhecido como "Regra de Hebb" ou "Lei de Hebb".

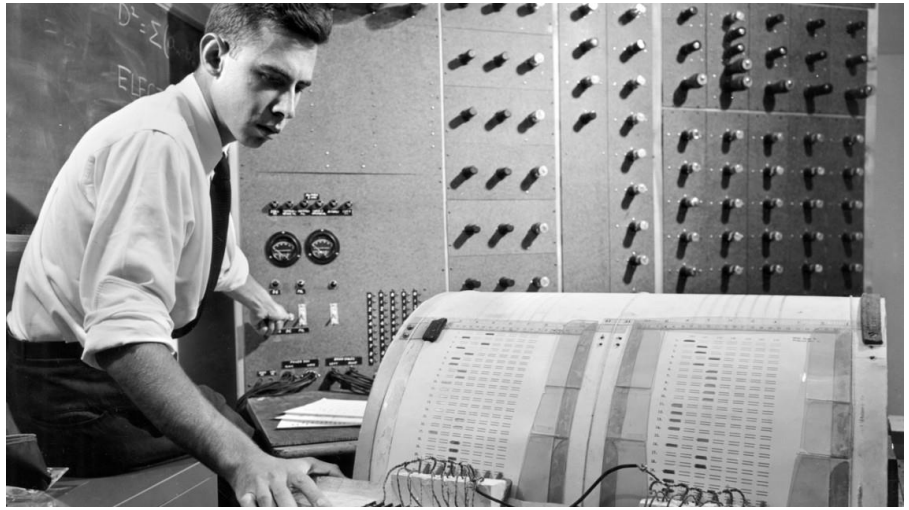
"Quando o axônio de uma célula A está perto o suficiente para excitar uma célula B e participa de sua ativação de forma repetitiva... a eficiência da célula A na ativação da B aumenta."

Neurons that fire together, wire together" (Neurônios que disparam juntos, conectam-se).

Primeiros desenvolvimentos

O Perceptron de Rosenblatt

Em 1957, o psicólogo Frank Rosenblatt, do *Cornell Aeronautical Laboratory*, introduziu o Perceptron.



O Perceptron foi um marco: ele pegou o neurônio limiar de McCulloch-Pitts e combinou-o com uma regra de aprendizado prática, criando o primeiro modelo que poderia **aprender** com os dados para resolver problemas.

O Perceptron de Rosenblatt

O Perceptron foi o primeiro modelo a introduzir algoritmicamente o conceito de **aprendizado de máquina**.Ele demonstrou que um algoritmo simples poderia adaptar seus parâmetros (os pesos sinápticos) com base nos dados de entrada e no erro de sua previsão, a fim de melhorar seu desempenho.

O anúncio do Perceptron gerou imensa excitação e controvérsia. Em 1958, o The New York Times relatou que a Marinha esperava que o Perceptron fosse "o embrião de um computador eletrônico que [a Marinha] espera que seja capaz de andar, falar, ver, escrever, se reproduzir e estar consciente de sua existência".

NEW NAVY DEVICE LEARNS BY DOING

Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

WASHINGTON, July 7 (UPI) —The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

Perceptron

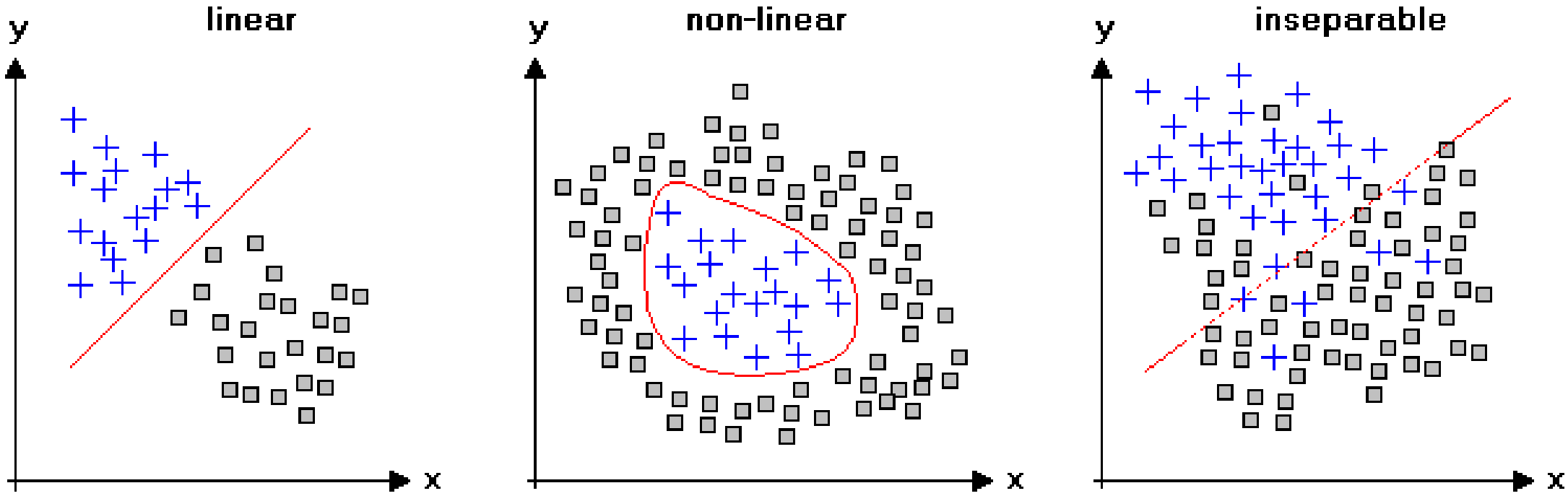
Perceptron

Em sua essência, o Perceptron de camada única é um **classificador linear binário**.

O modelo **calcula a saída a partir das entradas** da seguinte maneira: ele pega cada valor de entrada, multiplica pelo peso correspondente e soma todos esses resultados.

Em seguida, adiciona um valor extra chamado **bias**, que serve para ajustar o ponto de ativação do neurônio. O valor total obtido passa então por uma função — chamada **função de ativação** — que decide qual será a saída final do modelo.

Em outras palavras: o neurônio combina as entradas ponderadas pelos seus pesos, faz um pequeno ajuste com o bias e, por meio da função de ativação, transforma esse resultado em uma saída que pode ser usada para classificar, decidir ou estimar algo.


linear
y
x
non-linear
y
x
inseparable
y
x

A regra do aprendizado

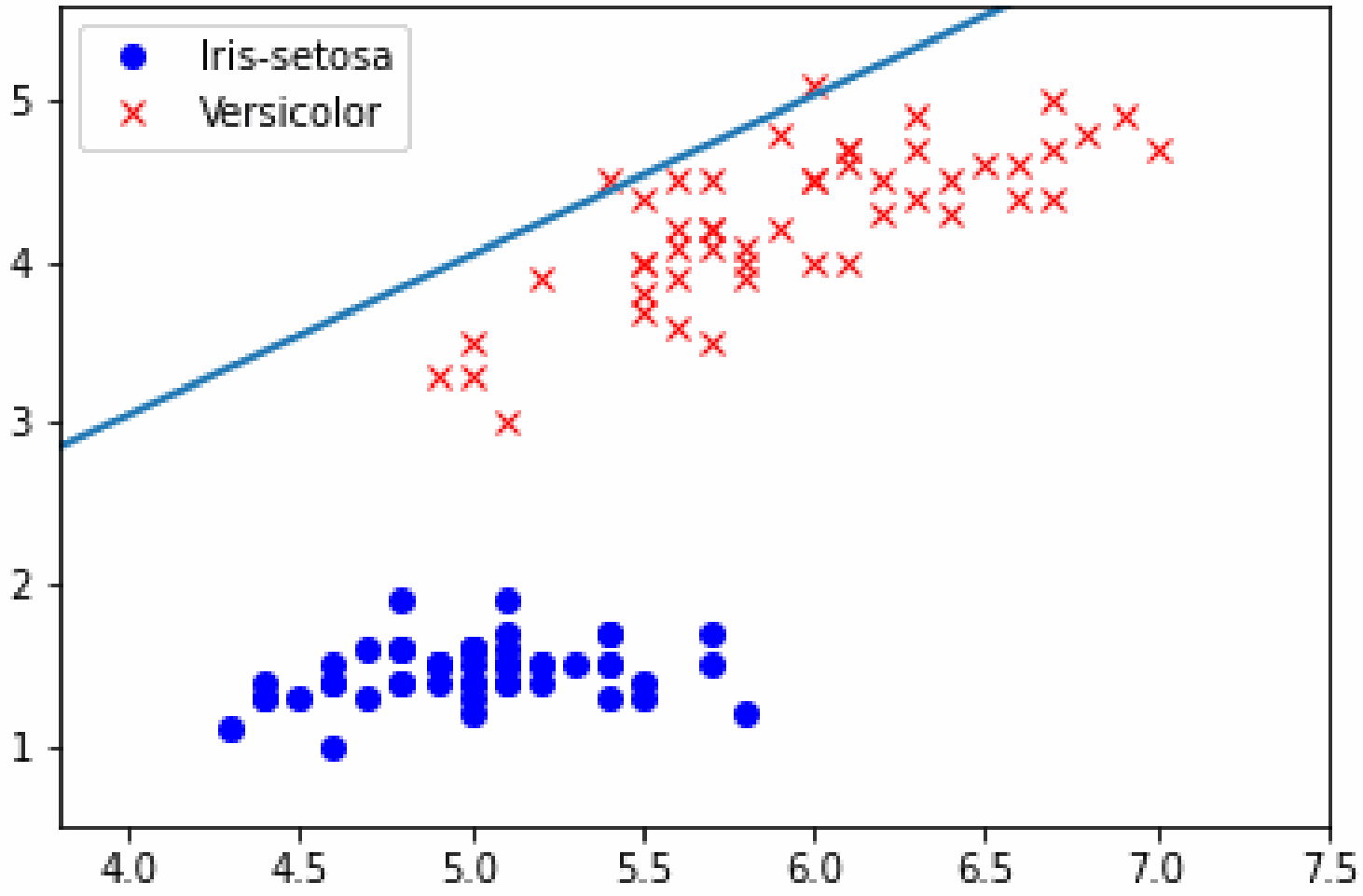
O Perceptron aprende por meio de um processo chamado **aprendizado supervisionado**, ou seja, ele recebe exemplos com respostas corretas já conhecidas. **Cada exemplo tem um conjunto de entradas e o rótulo que indica a classe correta.**

Durante o treinamento, o algoritmo segue um ciclo repetido de ajustes. Primeiro, os pesos são definidos com valores iniciais — normalmente zeros ou números pequenos aleatórios. Em seguida, o Perceptron calcula sua saída combinando as entradas com os pesos atuais. Essa saída é então comparada com o valor desejado, produzindo um erro, que mostra o quanto o modelo se afastou da resposta correta.

A regra do aprendizado

Se o resultado estiver certo, os pesos permanecem os mesmos. Caso contrário, eles são atualizados para reduzir o erro — movendo-se na direção que aproximará o resultado futuro do valor correto. A velocidade dessa correção é controlada por uma constante chamada taxa de aprendizado, que define o tamanho do ajuste em cada passo.

Esse processo se repete para todas as amostras do conjunto de treinamento, ajustando progressivamente os pesos até que o modelo consiga classificar corretamente (ou quase corretamente) todos os exemplos.


Iris-setosa
Versicolor
5
4
3
2
1
4.0
4.5
5.0
5.5
6.0
6.5
7.0
7.5

Limitação Fundamental: Separabilidade Linear

A garantia de convergência do Perceptron contém sua maior limitação: ela só se aplica se o problema for linearmente separável.

Um problema é linearmente separável se existir um único hiperplano que possa dividir perfeitamente o espaço de entrada, com todas as amostras da classe 1 de um lado e todas as amostras da classe 2 do outro.

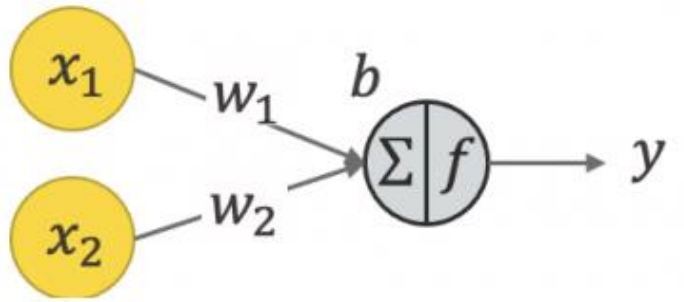
Conceito	Definição breve
Entrada (input)	Valor numérico que representa uma característica (feature) do exemplo analisado.
Peso (weight)	Coeficiente que indica a importância relativa de cada entrada no cálculo da saída do neurônio.
Bias (termo de polarização)	Valor adicionado à soma ponderada das entradas para ajustar o limiar de ativação.
Soma ponderada	Resultado da multiplicação de cada entrada pelo respectivo peso, seguido da soma de todos.
Função de ativação	Função que define se o neurônio “dispara” (ativa) ou não, transformando a soma ponderada em uma saída.
Saída (output)	Resultado final do neurônio, após aplicação da função de ativação.
Erro	Diferença entre a saída obtida e a saída desejada (esperada).
Taxa de aprendizado	Parâmetro que controla o tamanho do ajuste dos pesos a cada iteração do treinamento.
Época (epoch)	Passagem completa por todos os exemplos do conjunto de treinamento.
Regra de atualização dos pesos	Fórmula usada para corrigir os pesos com base no erro, normalmente $\Delta w = \eta \cdot erro \cdot x$.

Exemplo

Cenário

Nosso objetivo é classificar se uma pessoa é **Homem (1)** ou **Mulher (0)** com base em duas características (features):

- ***x1***: Altura (em metros)
- ***x2***: Peso (em dezenas de Kg)



Componentes Essenciais:

- **Entradas (Inputs):** *x1* (Altura) e *x2* (Peso).
- **Pesos (Weights):** *w1* (importância da altura) e *w2* (importância do peso).
- **Bias (Viés):** *b*. É um "limiar" (*threshold*) que ajusta a fronteira de decisão.

Cenário

- **Função de Soma (Net Input):** $z = (w1 \cdot x1) + (w2 \cdot x2) + b$
- **Função de Ativação (Degrau):**
    - Se z >= 0, a previsão (ŷ) é **1 (Homem)**.
    - Se z < 0, a previsão é (ŷ) **0 (Mulher)**.

- **Taxa de Aprendizado (η):** Define o "tamanho do passo" da correção. Usaremos η = **0.1**.

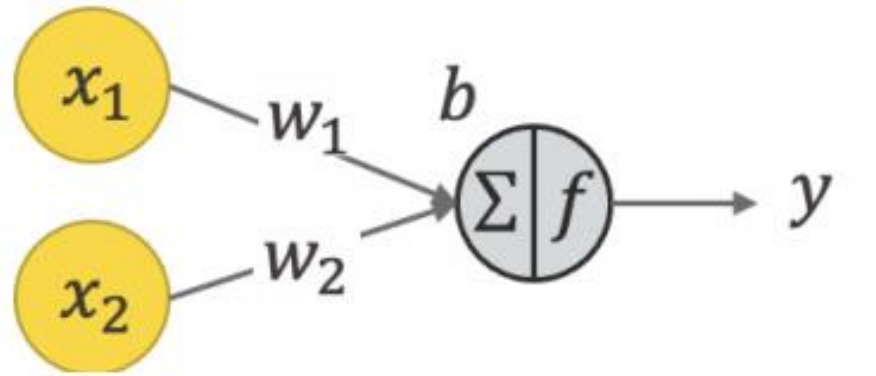
Cenário

Amostra	x1 (Altura)	x2 (Peso)	y (Esperado)
A	1.6	6.0	0 (Mulher)
B	1.7	6.5	0 (Mulher)
C	1.8	8.0	1 (Homem)
D	1.9	8.2	1 (Homem)

Estado Inicial (Época 0)

Vamos começar com valores pequenos e arbitrários para os pesos e o bias.

- w1 = 0.2
- w2 = 0.1
- b = -0.1
- η = 0.1



O Treinamento (Passo a Passo)

O treinamento ocorre em **"Épocas"**.

Uma época termina quando o modelo viu todos os dados de treino uma vez.
O modelo só aprende quando erra!

A Regra de Correção (quando ŷ é diferente de y):

- $Erro = y - \hat{y}$ (O esperado menos o previsto)
- $w_1 = w_1 + (\eta \cdot Erro \cdot x_1)$
- $w_2 = w_2 + (\eta \cdot Erro \cdot x_2)$
- $b = b + (\eta \cdot Erro)$

Época 1

Pesos Iniciais: $w_1 = 0.2$, $w_2 = 0.1$, $b = -0.1$

1. Amostra A: (1.6, 6.0), Esperado (y) = 0

- $z = (0.2 \cdot 1.6) + (0.1 \cdot 6.0) + (-0.1)$
- $z = 0.32 + 0.6 - 0.1 = 0.82$
- $z \geq 0$, então $\hat{y} = 1$.

Errou!

Correção:

- $Erro = 0 - 1 = -1$
- $w_1 = 0.2 + (0.1 \cdot -1 \cdot 1.6) = 0.2 - 0.16 = \mathbf{0.04}$
- $w_2 = 0.1 + (0.1 \cdot -1 \cdot 6.0) = 0.1 - 0.6 = \mathbf{-0.5}$
- $b = -0.1 + (0.1 \cdot -1) = -0.1 - 0.1 = \mathbf{-0.2}$

Época 1 – Amostra B

Usando os pesos atualizados: $w_1 = 0.04$, $w_2 = -0.5$, $b = -0.2$

$$z = (0.04 \cdot 1.7) + (-0.5 \cdot 6.5) + (-0.2)$$

$$z = 0.068 - 3.25 - 0.2 = -3.382$$

$z < 0$, então $\hat{y} = 0$.

Acertou! O que acontece com os pesos?

Época 1 – Amostra C

Pesos: $w_1 = 0.04$, $w_2 = -0.5$, $b = -0.2$

$$z = (0.04 \cdot 1.8) + (-0.5 \cdot 8.0) + (-0.2)$$

$$z = 0.072 - 4.0 - 0.2 = -4.128$$

$z < 0$, então $\hat{y} = 0$.

Errou ou acertou?

Como é feita a correção?

Continuem!