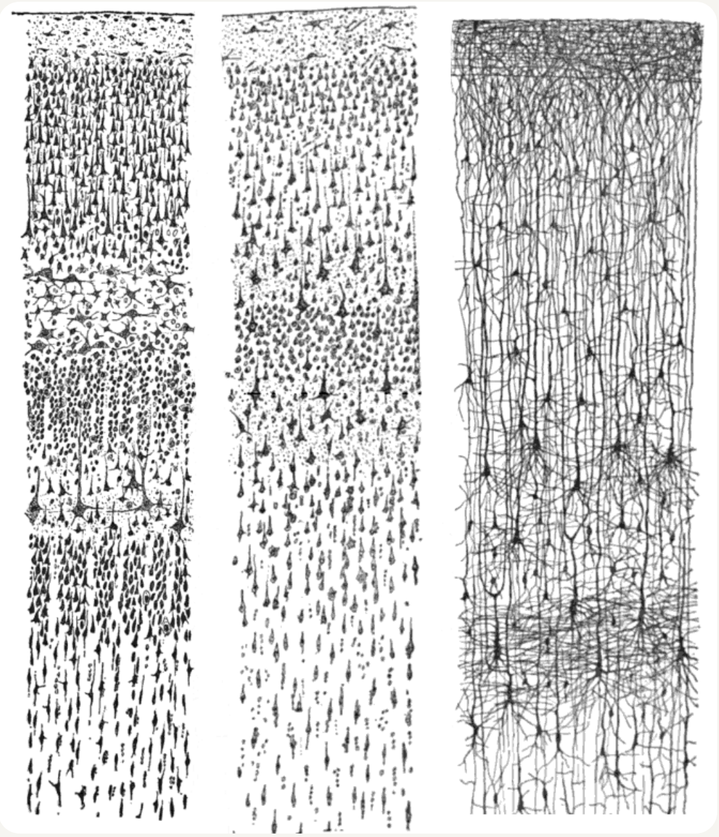
SIENA, NIPS-CERN, UFJF

Limiar

Redes neurais, do neurônio ao fractal

Aula 3: matrizes e camadas

13/11/2026. C∴ Chrysthofer A. A. Afonso



Camadas do córtex cerebral humano, em três desenhos de Santiago Ramón y Cajal, de 1899. Domínio público.

1 · REVISÃO

O que ficou das aulas 1 e 2, de memória

1. O que um neurônio calcula, e o que ele desenha no plano?

2. Por que nenhum neurônio sozinho acerta o XOR?

3. O que a perda mede?

4. O que o gradiente descendente faz a cada passo?

1 · REVISÃO

Das aulas 1 e 2, três contas

O neurônio

a soma ponderada mais o viés; o degrau decide

$$y = \text{degrau}(\vec{w} \cdot \vec{x} + b)$$

A perda

a média dos quadrados dos erros

$$L = \frac{1}{N} \sum_{i=1}^{N} (y_i - t_i)^2$$

O passo

cada parâmetro anda contra a própria derivada

$$w \leftarrow w - \eta \frac{\partial L}{\partial w}$$

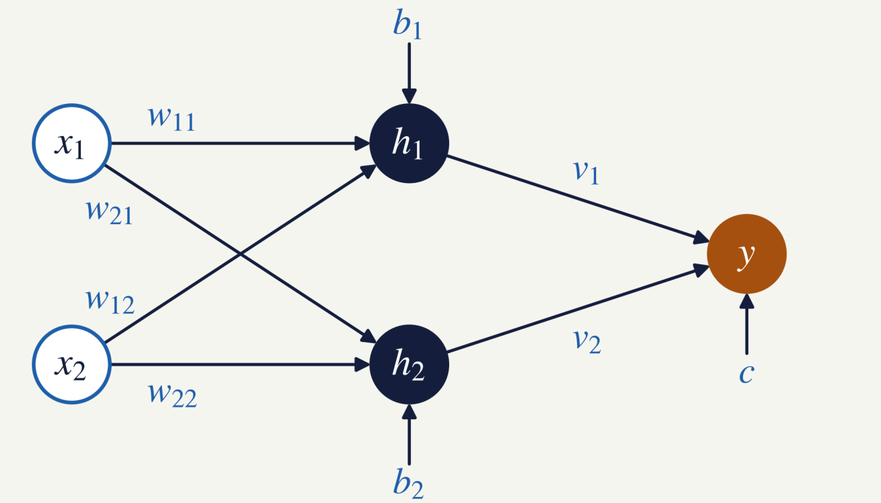
1 · REVISÃO

O fio do curso, da aula 1 à 10

1	**O neurônio e o perceptron**	um neurônio desenha uma reta, e nenhuma reta separa o XOR	25/09, presencial
2	**Derivada e gradiente**	o passo contra a derivada; a taxa grande demais faz o treino divergir	02/10, presencial
3	**Matrizes e camadas**	duas camadas separam o XOR, com pesos escolhidos à mão	13/11, on-line
4	**O Jacobiano**	a regra da cadeia com vetores, uma camada de cada vez	13/11, on-line
5	**Retropropagação**	a regra da cadeia, de trás para a frente, acha os pesos do XOR	13/11, on-line
6	**O MNIST em NumPy**	um MLP em 70 mil dígitos escritos à mão, com treino, validação e teste	27/11, on-line
7	**PyTorch**	a biblioteca faz a retropropagação; o erro final contra a taxa	27/11, on-line
8	**Os dados do CERN**	um MLP estima a massa dos dois múons, conferida pela fórmula	04/12, on-line
9	**O fractal e o projeto**	a fronteira entre a taxa que converge e a que diverge	04/12, on-line
10	**A apresentação**	a apresentação do projeto e a revisão do curso inteiro	11/12, on-line

FAPEMIG CNPq Conselho Nacional de Desenvolvimento Científico e Tecnológico CAPES

Uma camada é uma conta só



CAMADA OCULTA

$$\vec{h} = \text{degrau}(W\vec{x} + \vec{b})$$

o vetor h é o degrau, posição por posição, de *W* vezes o vetor x, mais o vetor b

$$W\vec{x} + \vec{b} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} w_{11}x_1 + w_{12}x_2 + b_1 \\ w_{21}x_1 + w_{22}x_2 + b_2 \end{pmatrix}$$

CAMADA DE SAÍDA

$$y = \text{degrau}(\vec{v} \cdot \vec{h} + c)$$

Matriz, no curso: uma tabela de números com *m* linhas e *n* colunas; aqui, a linha *k* de *W* guarda os pesos do neurônio *k*.

A matriz vezes o vetor, número a número

2 por 2 vezes 2 números dá 2 números, um por neurônio

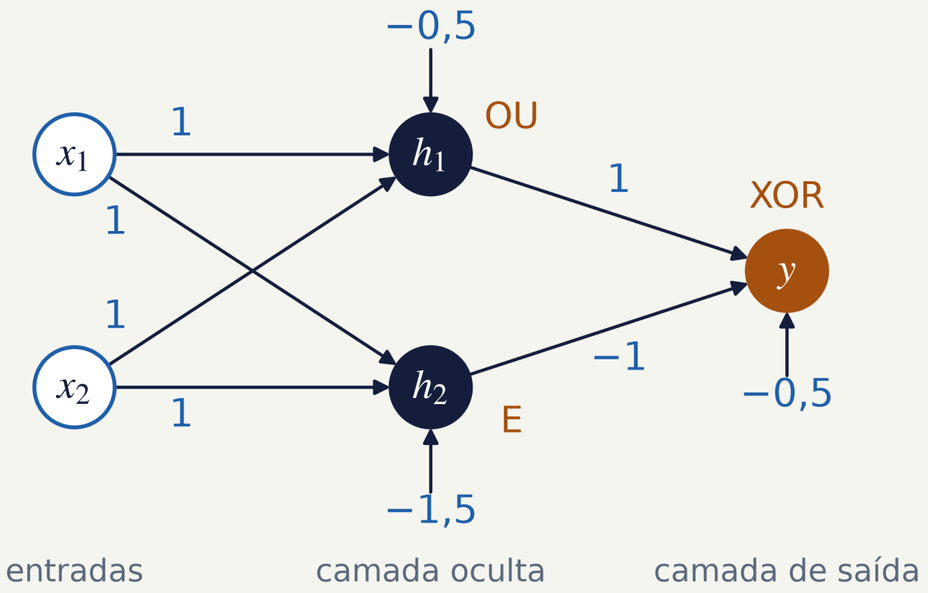
$$\underset{\text{2 por 2}}{\begin{pmatrix} 1 & -2 \\ 3 & 0{,}5 \end{pmatrix}} \underset{2}{\begin{pmatrix} 2 \\ 1 \end{pmatrix}} + \underset{2}{\begin{pmatrix} 0{,}5 \\ -1 \end{pmatrix}} = \underset{2}{\begin{pmatrix} 0{,}5 \\ 5{,}5 \end{pmatrix}}$$

$$1 \cdot 2 + (-2) \cdot 1 + 0{,}5 = 0{,}5$$

$$3 \cdot 2 + 0{,}5 \cdot 1 + (-1) = 5{,}5$$

$$\vec{h} = \text{degrau}(\vec{z}) = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

O XOR com pesos escolhidos à mão



x_1	x_2	h_1	h_2	y
		OU	E	XOR
0	0	0	0	0
0	1	1	0	**1**
1	0	1	0	**1**
1	1	1	1	0

y dispara só com $h_1 = 1$ e $h_2 = 0$: o OU sem o E, que é o XOR.

As quatro entradas numa conta só

$$Z = X\,W^{\top} + \vec{b}$$

W transposta: as linhas de *W* viram colunas; cada linha de *X* é uma entrada

$$\begin{pmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} + \begin{pmatrix} -0{,}5 & -1{,}5 \end{pmatrix} = \begin{pmatrix} -0{,}5 & -1{,}5 \\ 0{,}5 & -0{,}5 \\ 0{,}5 & -0{,}5 \\ 1{,}5 & 0{,}5 \end{pmatrix}$$

$$H = \text{degrau}(Z) = \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 1 & 1 \end{pmatrix}$$

LOTE

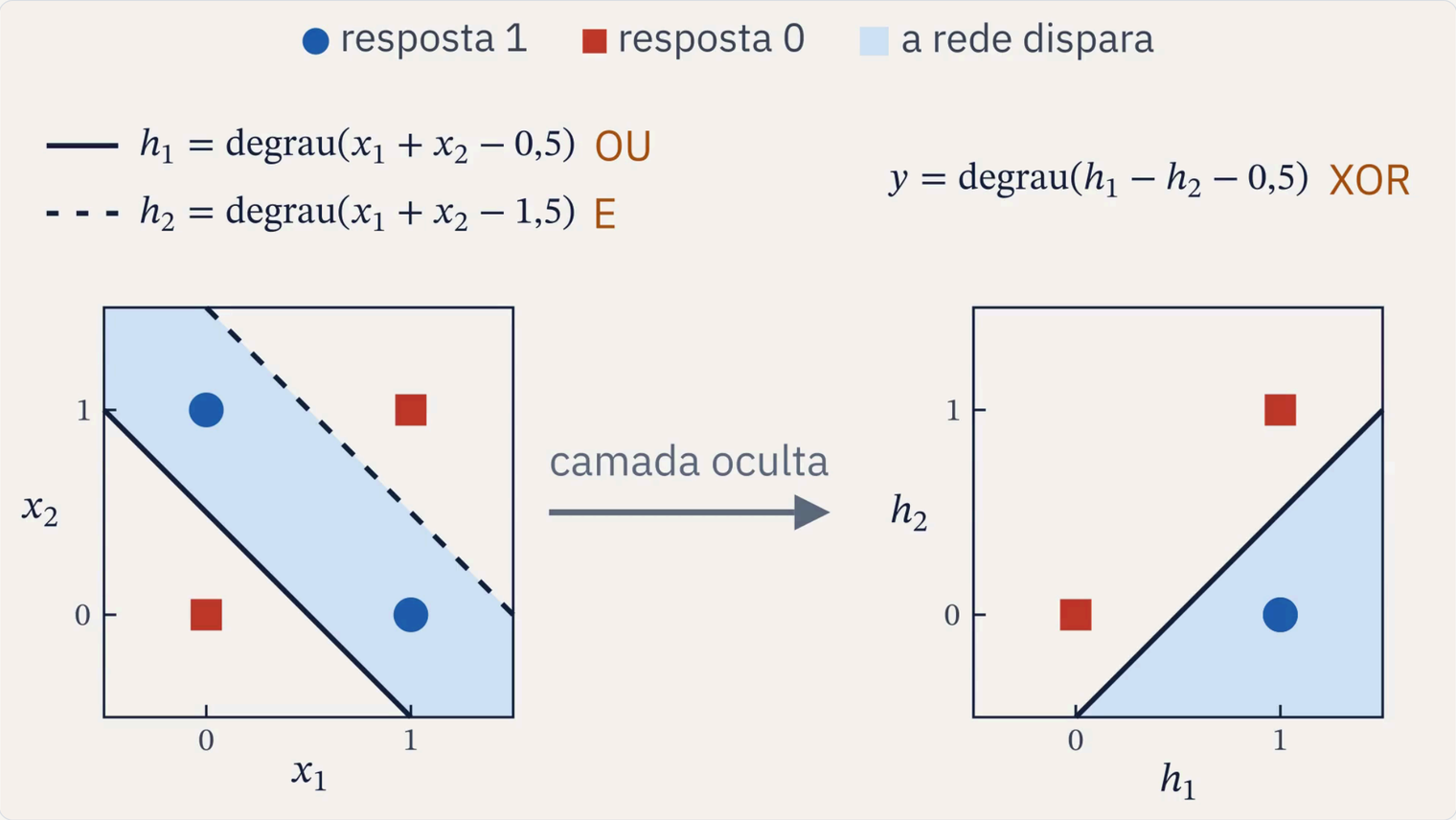
as entradas que passam juntas pela rede, uma por linha de X; aqui, as quatro do XOR

NO NUMPY

```
X @ W.T + b
```

o NumPy soma b a cada linha; é a forma do código da aula 6, com 64 imagens por lote

A camada oculta muda o plano



Camada, camada oculta e MLP

Camada	neurônios que leem as mesmas entradas, cada um com os seus pesos; a saída de uma camada é a entrada da seguinte
Camada oculta	a que fica entre a entrada e a saída: não recebe os dados nem dá a resposta. O artigo de 1986 de Rumelhart, Hinton e Williams chama de hidden as unidades que não são entrada nem saída
Perceptron multicamadas, o MLP	camadas empilhadas, cada neurônio ligado a todos os da camada seguinte. A rede do XOR é um MLP 2-2-1: duas entradas, dois neurônios ocultos, uma saída

Quantos números a rede guarda

$$n \cdot m + m$$

uma camada de n entradas e m neurônios: os pesos de W e um viés por neurônio

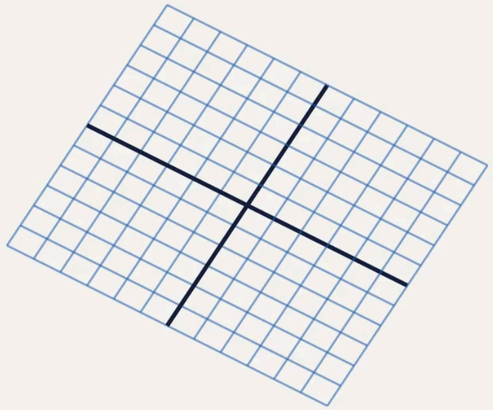
a rede	as camadas	os parâmetros
o XOR, aulas 3 e 5	2, 2, 1	**9**
a rede pequena, aulas 7 e 9	784, 16, 10	**12.730**
o MNIST, aula 6	784, 256, 10	**203.530**
a massa invariante, aula 8	8, 256, 256, 1	**68.353**

Na rede do MNIST, a primeira camada guarda 200.960 dos 203.530: cada pixel liga a cada neurônio.

O plano inteiro: matrizes e a sigmoide

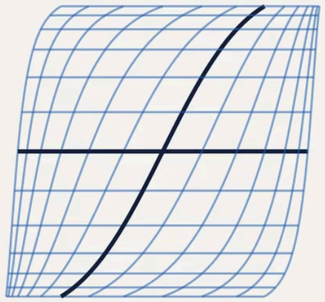
Com a sigmoide em cada coordenada, as retas entortam

$$W_2 W_1 = \begin{pmatrix} 1 & 0{,}5 \\ -0{,}5 & 0{,}75 \end{pmatrix}$$



sem ativação: retas

$$\sigma(W_1 \vec{x})$$



com a sigmoide: curvas, de 0 a 1

3 · A ATIVAÇÃO

Sem ativação, duas camadas valem uma

$$W_2\,(W_1\vec{x} + \vec{b}_1) + \vec{b}_2 = (W_2 W_1)\,\vec{x} + (W_2\vec{b}_1 + \vec{b}_2)$$

W_2W_1, W dois vezes W um: o produto de duas matrizes, W_1 e depois W_2 numa matriz só

A forma é a de uma camada só, W vezes o vetor x mais um vetor: de novo uma reta por neurônio de saída, e o XOR sem solução.

CONFERIDO EM NUMPY

$$\text{diferença} = 7 \cdot 10^{-15}$$

com matrizes sorteadas, semente 3: o arredondamento do computador

POR ISSO

a camada oculta precisa de uma ativação que não seja linear, como o degrau

Duas camadas lineares, com números

	pesos	viés
1ª camada	$W_1 = \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix}$	$\vec{b}_1 = \begin{pmatrix} 1 \\ -1 \end{pmatrix}$
2ª camada	$W_2 = \begin{pmatrix} 1 & -1 \end{pmatrix}$	$b_2 = 3$

A CAMADA ÚNICA

$$W_2 W_1 = \begin{pmatrix} 1 & -1 \end{pmatrix} \begin{pmatrix} 1 & 2 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 \end{pmatrix}$$

$$W_2 \vec{b}_1 + b_2 = 1 \cdot 1 + (-1)(-1) + 3 = 5$$

COM O VETOR X = (2; 1)

pelas duas camadas

$$W_1 \vec{x} + \vec{b}_1 = (5;\ 0)$$

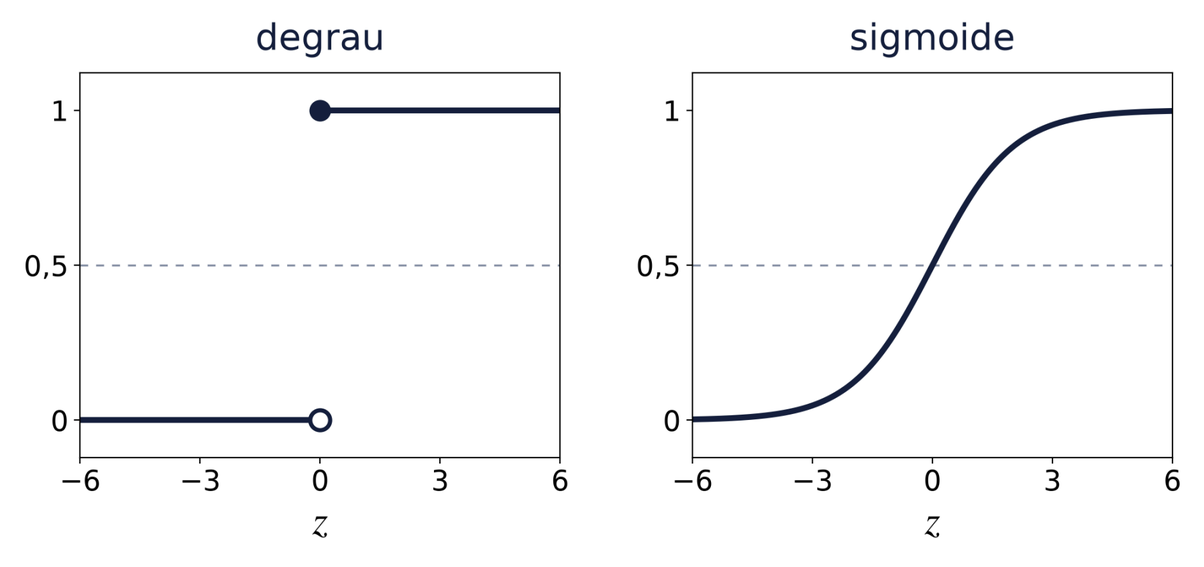
$$W_2\,(5;\ 0) + 3 = 8$$

pela camada única

$$1 \cdot 2 + 1 \cdot 1 + 5 = 8$$

3 · A ATIVAÇÃO

Do degrau à sigmoide



$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

sigma de z é 1 sobre 1 mais e elevado a menos z; $e \approx 2{,}718$, o número de Euler

$$\sigma'(z) = \sigma(z)\,(1 - \sigma(z))$$

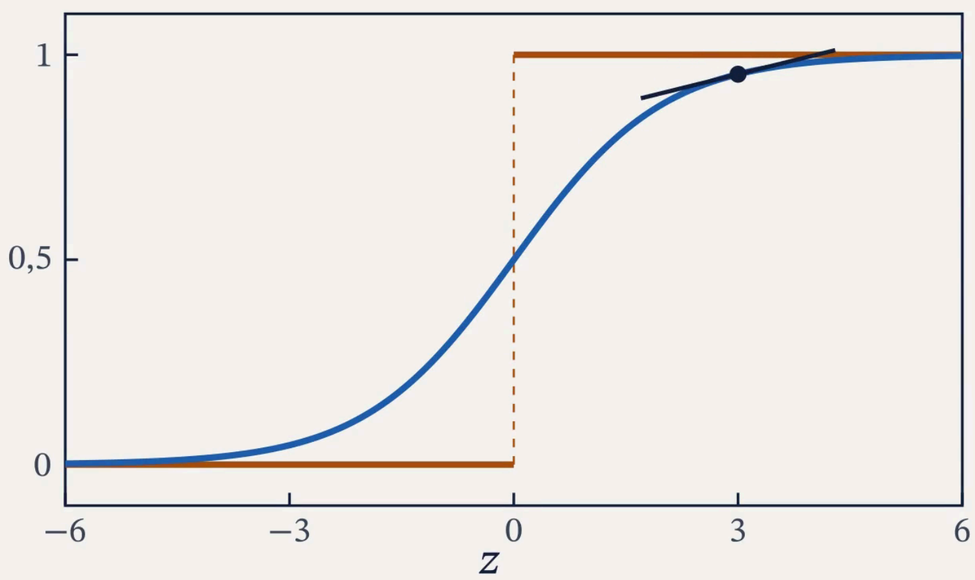
sigma linha de z: a inclinação da sigmoide

O degrau é plano fora do zero: a derivada é zero, e o gradiente não anda.

A sigmoide é lisa: tem inclinação em todo ponto, e a saída fica entre 0 e 1.

A sigmoide vira degrau, e o degrau não inclina

Inclinação zero, passo zero: o degrau não treina



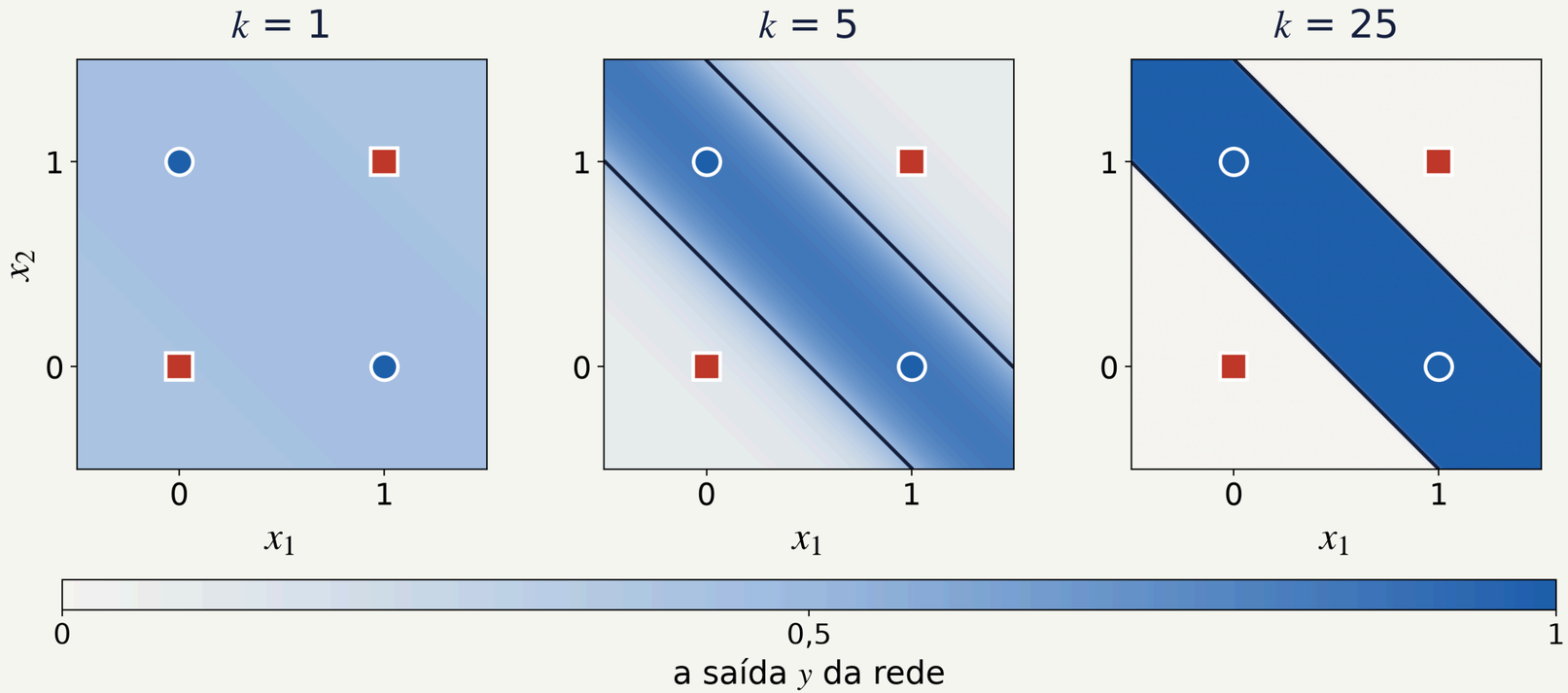
$\sigma(kz)$

$k = 1{,}0$

no ponto $z = 3{,}0$

inclinação $0{,}045$

Os pesos do XOR com a sigmoide



Pesos à mão vezes k: com $k = 1$, a rede erra (0; 1) e (1; 0); com $k = 5$, acerta; com $k = 25$, é o degrau.

3 · A ativação

Como achar os nove números da rede sem escolher à mão?

A resposta é a aula 5: a retropropagação.

4 · O CÓDIGO

A rede do XOR em NumPy

1. A rede como lista de camadas, cada uma a matriz *W* e o vetor dos vieses
2. A passada para a frente, para qualquer número de camadas: a saída de uma é a entrada da seguinte
3. A tabela das quatro entradas, conferida contra o XOR
4. Duas camadas lineares contra uma, com matrizes sorteadas

O NUMPY QUE RESOLVE

```
W @ x + b
(z >= 0).astype(int)
W2 @ W1
np.abs(a - b).max()
```

o @ é o produto de matrizes; o astype troca True e False por 1 e 0

```
python aulas/03-camadas/codigo/xor_a_mao.py
```

5 · FECHAMENTO

Para estudar e entregar

Estudo

- 3Blue1Brown, cálculo, capítulo 4: a regra da cadeia
- Khan Academy: o Jacobiano, três vídeos

Entrega

- a passada para a frente para qualquer número de camadas
- duas camadas lineares contra uma, em NumPy
- uma pergunta sobre o estudo

A seguir

- aula 4: a regra da cadeia com vetores, o Jacobiano
- aula 5: a retropropagação acha os nove pesos