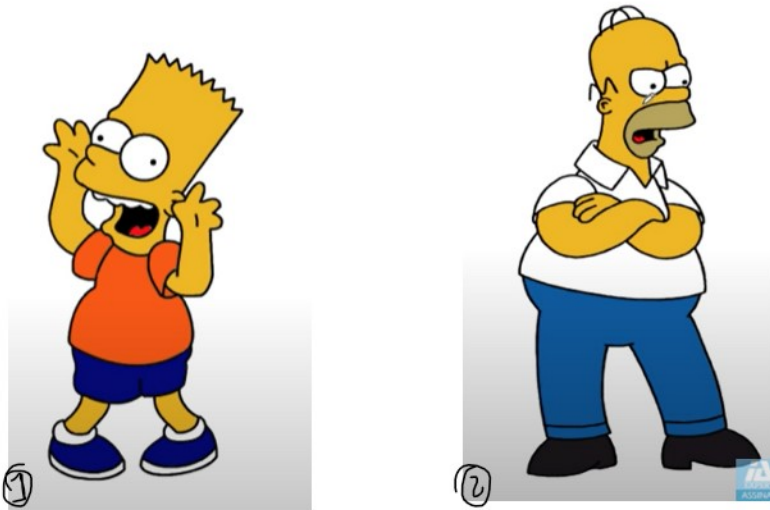


# Redes Neurais convolucionais

Classificação:



Inicialmente penso fazer uma extração de características.

Por exemplo:

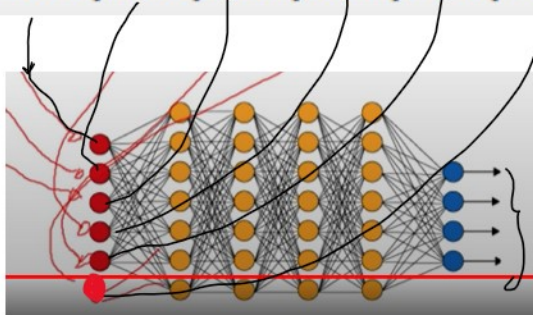
1)

- Laranja camisa
- Azul calção
- Azul sapato

2)

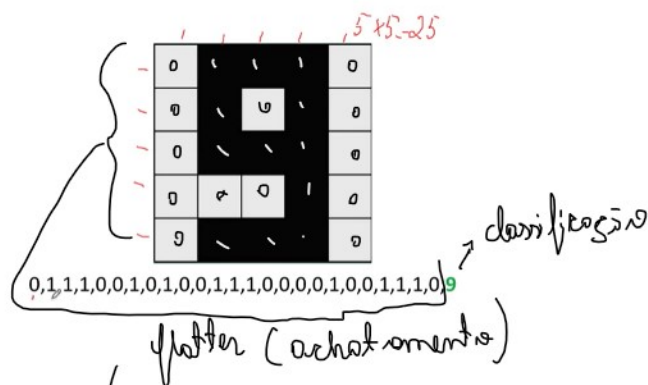
- Marrom boca
- Azul calca
- Cinza sapato

8.97, 3.45, 2.35, 0.0, 00.00, 0.00, Bart  
 6.75, 0.94, 0.52, 0.00, 0.00, 0.00, Bart  
 9.69, 4.10, 1.56, 0.00, 0.00, 0.00, Bart  
 0.00, 0.00, 0.00, 4.68, 0.66, 0.01, Homer  
 0.00, 0.00, 0.00, 0.12, 2.50, 0.03, Homer  
 0.00, 0.00, 0.00, 5.80, 0.50, 1.28, Homer



→ Provavelmente seria só 2 nos  
 14 nos duas classificações

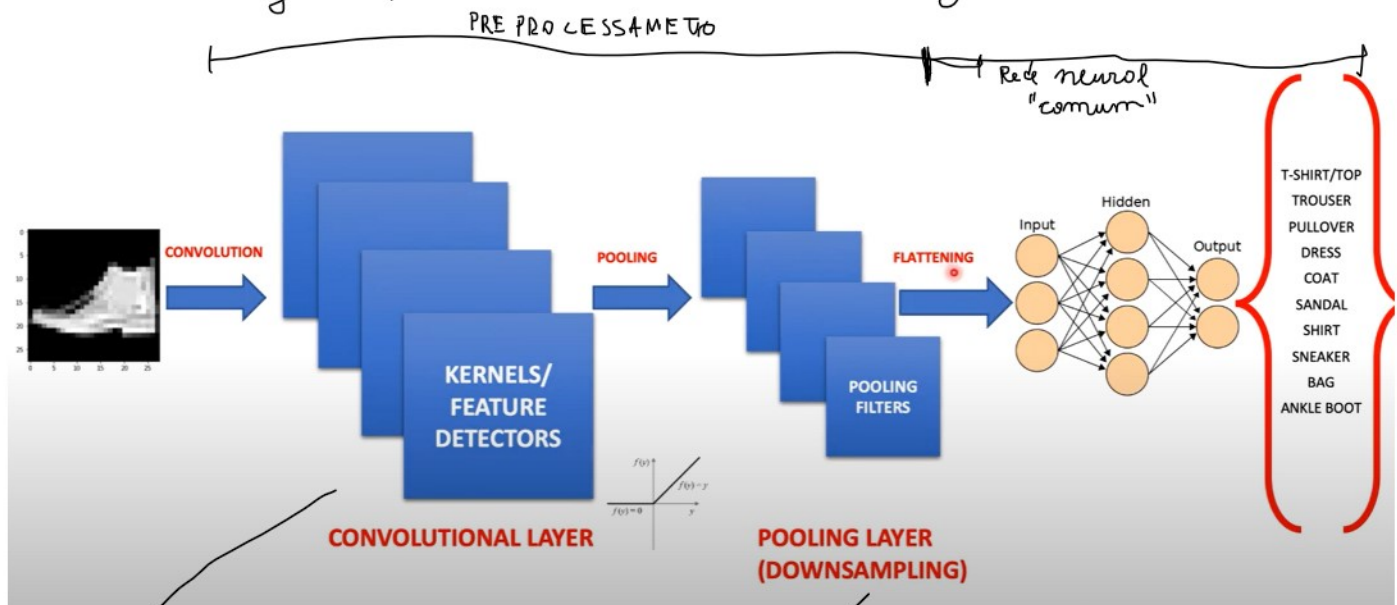
Outro problema de classificação



Como de entrada vai ter 25 mas a  
como de saída terá 10

O que difere uma rede neural normal para uma CNN?

Antes terá alguns processamento com as imagens



Reduz as dimensões







## DETECTORES DE CARACTERÍSTICAS

- As convoluções usam uma matriz para varrerem a imagem e aplicar um filtro para obter certo efeito
- Kernel é uma matriz para aplicar efeitos como embaçamento, por exemplo
- Selecionam as características mais importantes da imagem (pixels mais importantes)
- As convoluções preservam a relação espacial entre os pixels

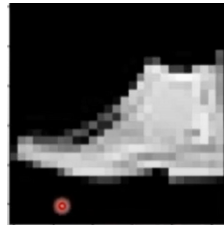
Exemplos de efeito

Operation	Kernel $w$	Image result $g(x_i)$
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	

esses efeitos ajudam a selecionar os pixels

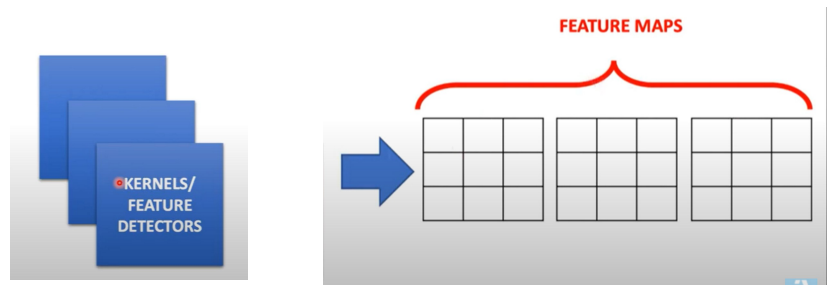
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & 1 & -1 \\ -1 & 8 & -1 \\ -1 & 1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	

esses efeitos ajudam a selecionar os pixels + importantes. Por exemplo, na figura, houve um



filtro que remove o fundo, logo manteve os pixels + importantes.

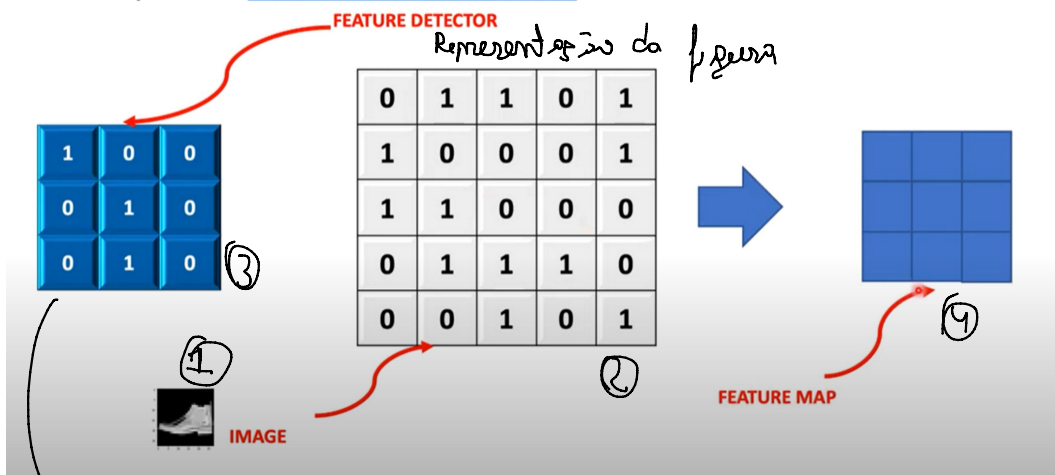
Podem ter vários filtros/kernel



3 kernels

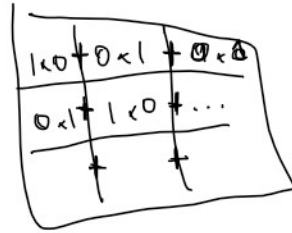
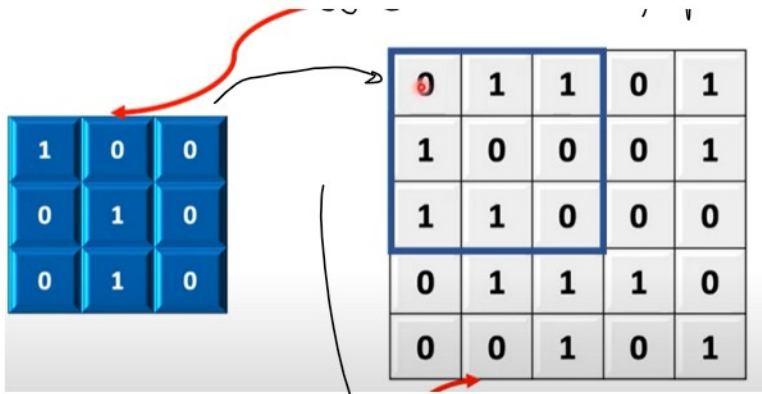
3 mapa de caract.

- Exemplo on-line: <http://setosa.io/ev/image-kernels/>



o filtro. A gente pode escolher o filtro e qual a dim  
Logo a gente aplica 3 em 2 e gera o 4 (mapa de caract)  
detecção de caract / filtro / kernel





Resultado em  
1

2 vari andon do na

matriz

Resultado em 1

0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
0	1	1	1	0
0	0	1	0	1

...

0	1	1	0	1
1	0	0	0	1
1	1	0	0	0
0	1	1	1	0
0	0	1	0	1

...

1	1	1
3	1	1
2	3	1

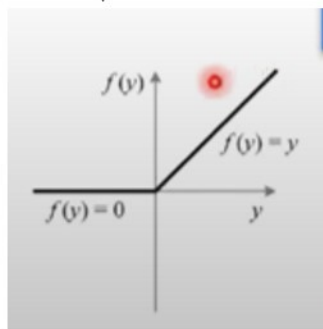
mapa de conclusões

# Rolling

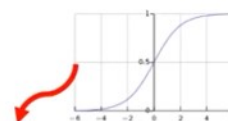
- Usado para adicionar não linearidade no mapa de características
- Aprimora a dispersão do mapa de características



Linha feita com ReLU



## RELU (RECTIFIED LINEAR UNITS)



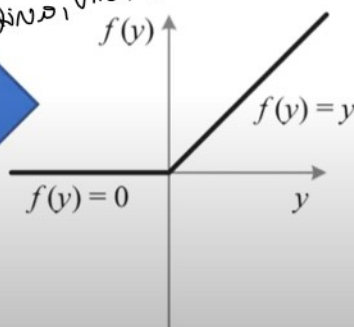
- O gradiente não desaparece se comparado com a função sigmoide

Mapa de características

7	10	-5	2	1
1	0	2	3	-6
1	17	-5	0	0
0	1	1	1	0
0	0	-8	12	1

onde é negativo, vira zero

ReLU



7	10	0	2	1
1	0	2	3	0
1	17	0	0	0
0	1	1	1	0
0	0	0	12	1



# POOLING (DOWNSAMPLING)

- Reduz a dimensionalidade do mapa de características
- Aumenta a eficiência computacional, preservando as características
- Ajuda o modelo a generalizar melhor, prevenindo o overfitting
- Se um pixel muda de lugar, o mapa será o mesmo

1	1	3	4
3	6	2	8
3	9	1	0
1	3	3	4

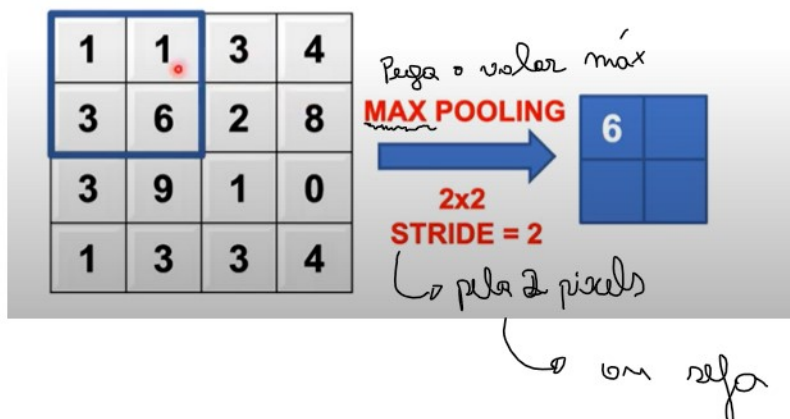
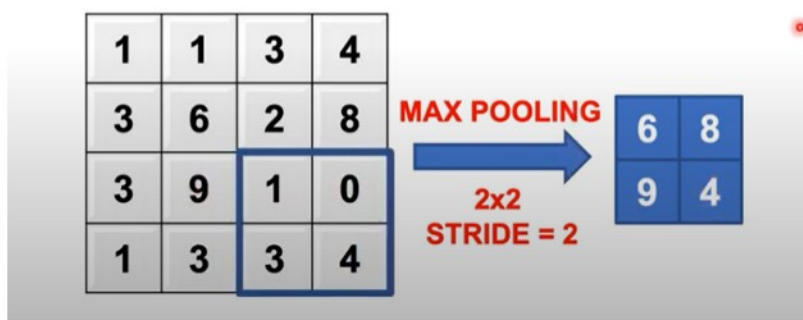


Diagram illustrating the second step of MAX POOLING. The 4x4 input grid is shown with a 2x2 region highlighted (top-right). The value 3 is marked as the maximum in this region. A blue arrow points to a 2x2 output grid where the top-right cell contains the value 3. Handwritten notes include: "Pega o valor máx" (Takes the max value), "MAX POOLING", "2x2 STRIDE = 2", and "pula 2 pixels".

1 1 3 4  
3 6 2 8  
3 9 1 0  
1 3 3 4

Pega o valor máx  
**MAX POOLING**  
2x2  
STRIDE = 2  
pula 2 pixels

3



## FLATTENING

