

CNN

Redes Neurais Convolucionais

Prof. Luiz Bordignon

Redes neurais convolucionais (CNN)

- Uma das áreas mais comuns de aplicação para CNN's é visão computacional.
- Identificação de objetos, reconhecimento fácil, robôs, carros, etc...

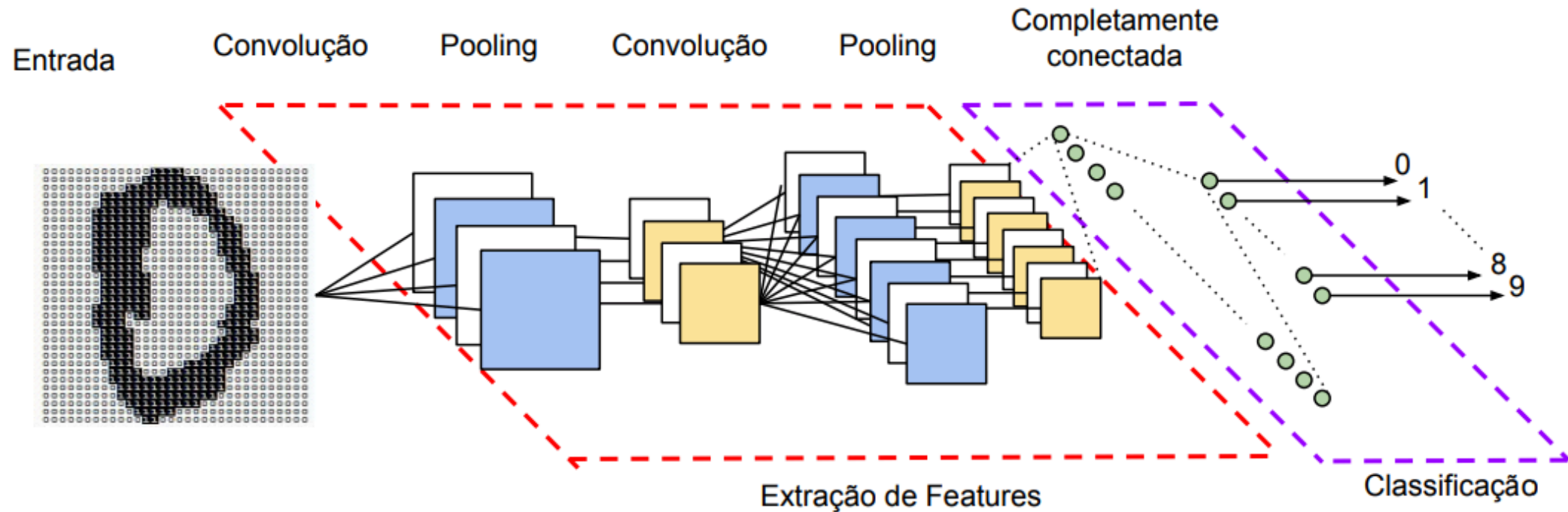
Redes neurais convolucionais (CNN)

- Quando usar CNN?
- Em problemas em que existam características complexas para diferenciar uma classe de outra.
- A CNN irá encontrar de forma automática, quais características são mais relevantes para resolução do problema.

Redes neurais convolucionais (CNN)

- Uma Rede Neural Convolucional (Convolutional Neural Network - CNN) é uma variação das redes de Perceptrons de Múltiplas Camadas.
- Uma rede convolucional utiliza além de uma rede neural densa, métodos de pré-processamento dos dados.
- Um desse processo é chamado de convolução, que é a aplicação de filtros/kernels nos dados.
- Outro processo utilizado é chamado de *polling*, que destaca os elementos mais significativos desse dados.

Exemplo visual de CNN

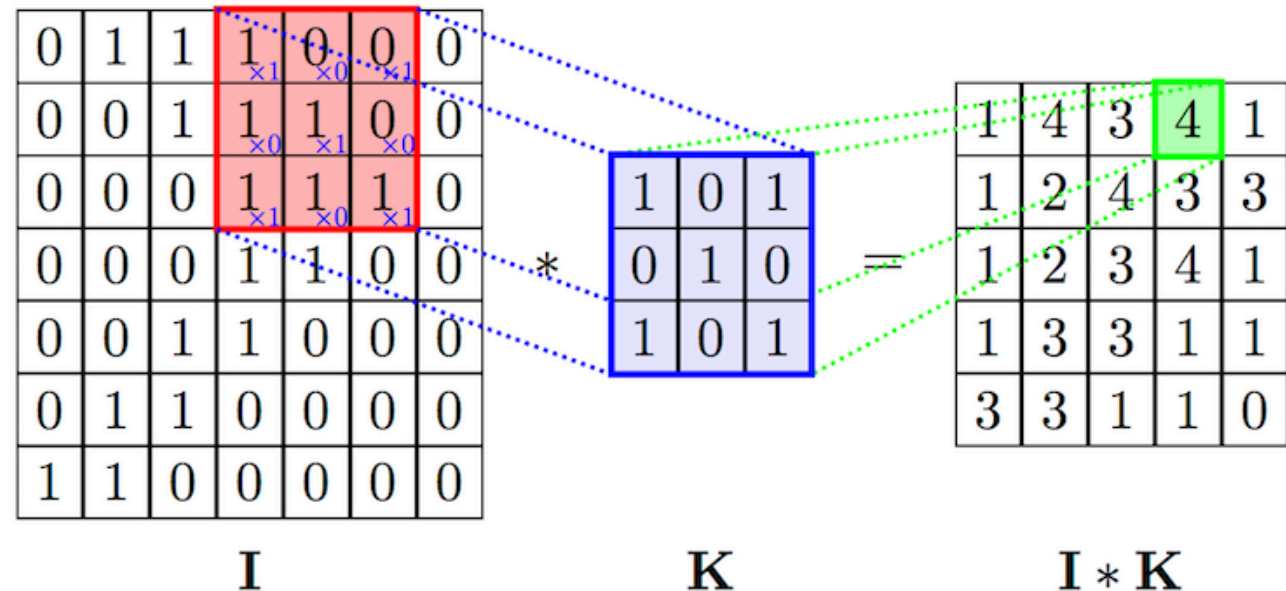


Redes neurais convolucionais (CNN)

- Podemos separar a fase inicial de uma CNN em 3 partes principais
- Convolução
- Polling
- Flattening

Convolução

- De forma bem simplificada, convolução é a multiplicação de uma matriz a um dado inicial.
- Podemos definir essa matriz como um filtro.
- Esse filtro pode ter seu tamanho definido, assim como a vizinhança que ele irá atuar (*stride*).



Convolução

↓

49

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

Imagem

x

3 x 3

1	0	0
1	0	1
0	1	1

Detector de características
(feature detector)

=

25

0				

Mapa de características
(feature map)

$$0 * 1 + 0 * 0 + 0 * 0 + 0 * 1 + 1 * 0 + 0 * 1 + 0 * 0 + 0 * 1 + 0 * 1 = 0$$

Convolução

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

Imagem

\times

1	0	0
1	0	1
0	1	1

Detector de características
(feature detector)

$=$

0	1			

Mapa de características
(feature map)

$$0 * 1 + 0 * 0 + 0 * 0 + 1 * 1 + 0 * 0 + 0 * 1 + 0 * 0 + 0 * 1 + 0 * 1 = 1$$

Convolução

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

Imagem

\times

1	0	0
1	0	1
0	1	1

Detector de características
(feature detector)

$=$

0	1	0		

Mapa de características
(feature map)

$$0 * 1 + 0 * 0 + 0 * 0 + 0 * 1 + 0 * 0 + 0 * 1 + 0 * 0 + 0 * 1 + 0 * 1 = 0$$

Convolução

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

Imagem

\times

1	0	0
1	0	1
0	1	1

Detector de características
(feature detector)

=

0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

Mapa de características
(feature map)

$$1 * 1 + 0 * 0 + 0 * 0 + 1 * 1 + 0 * 0 + 1 * 1 + 0 * 0 + 1 * 1 + 1 * 1 = 5$$

Aplicação de filtros

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

Convolução

- Dependendo do tamanho da matriz do filtro aplicado a imagem (dados) pode ter sua dimensionalidade reduzida.
- Facilita o processamento.
- Destaca características mais importantes, e elimina características menos significativas.

Convolução

- É comum, logo após a convolução, aplicar uma função de ativação.

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

Imagem

\times

1	0	0
1	0	1
0	1	1

Detector de características
(feature detector)

=

0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

Mapa de características
(feature map)



Camada de convolução

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

Imagem



utilizado

0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

Mapas de características (feature maps)

A rede decidirá qual detector de características que será utilizado

Camada de convolução é o conjunto de mapa de características

Pooling

- Outra camada muito importante comumente utilizada após as camadas de convolução e ativação, é a camada de agrupamento (pooling).
- A função dessa camada é reduzir a dimensionalidade dos dados na rede. Essa redução é importante por questão de agilidade no treinamento.
- A camada de pooling funciona agrupando um conjunto de dados, por exemplo: a entrada é dividida em janelas 4x4 e de cada uma é selecionado um valor para os representar.
- Essa escolha pode ser feita por diversas funções, porém a mais utilizada é a função de máximo.

Polling

0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

Mapa de características
(feature map)



2		

0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

Mapa de características
(feature map)



2	1	

0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

Mapa de características
(feature map)



2	1	2

0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

Mapa de características
(feature map)



2	1	2
3	3	2
3	3	

0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

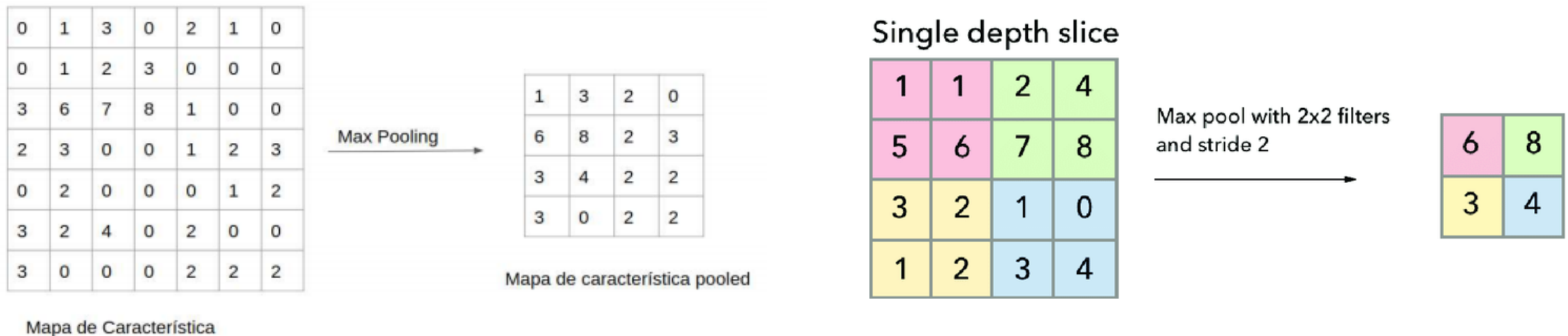
Mapa de características
(feature map)



2	1	2
3	3	2
3	3	5

Pooling

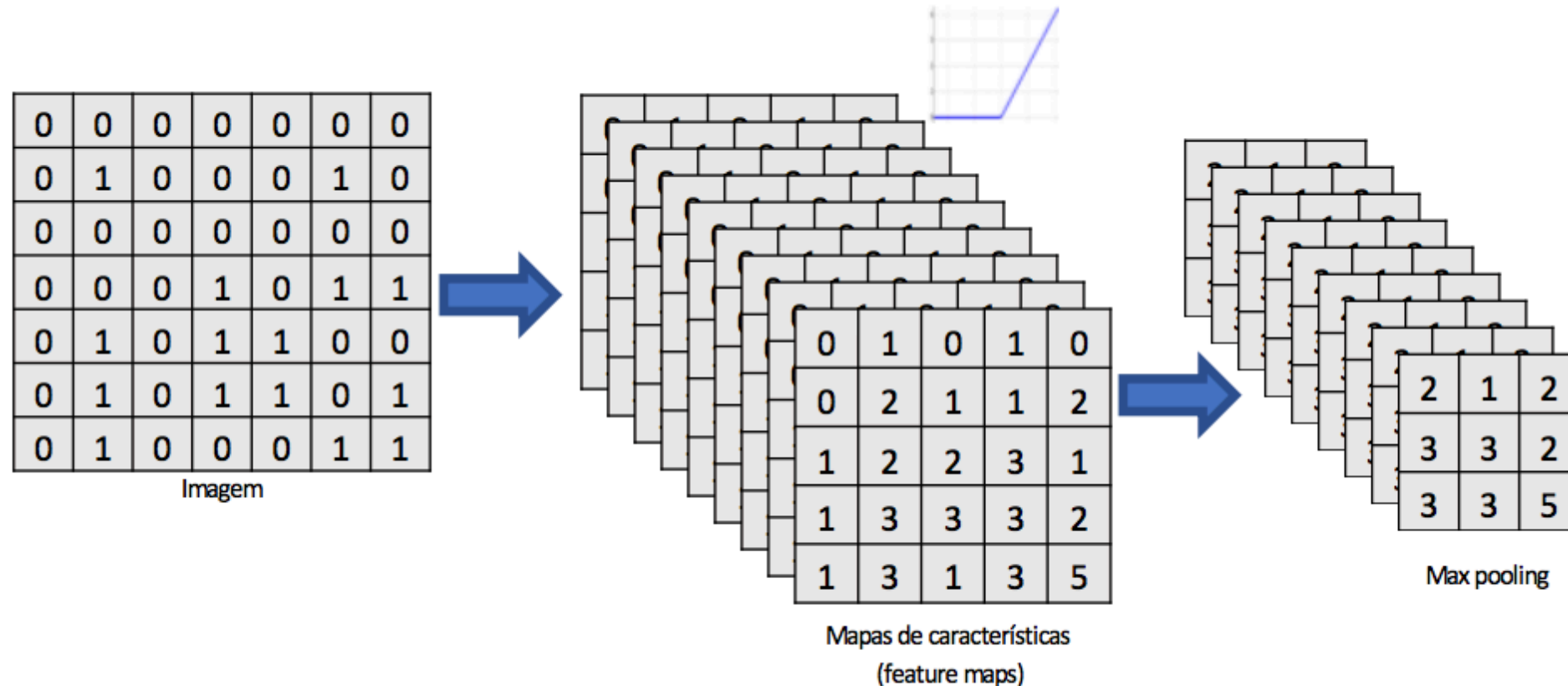
- Seleciona as características mais relevantes (reduz overfitting e ruídos desnecessários)
- Max pooling (mínimo, média): max foca nas características mais relevantes



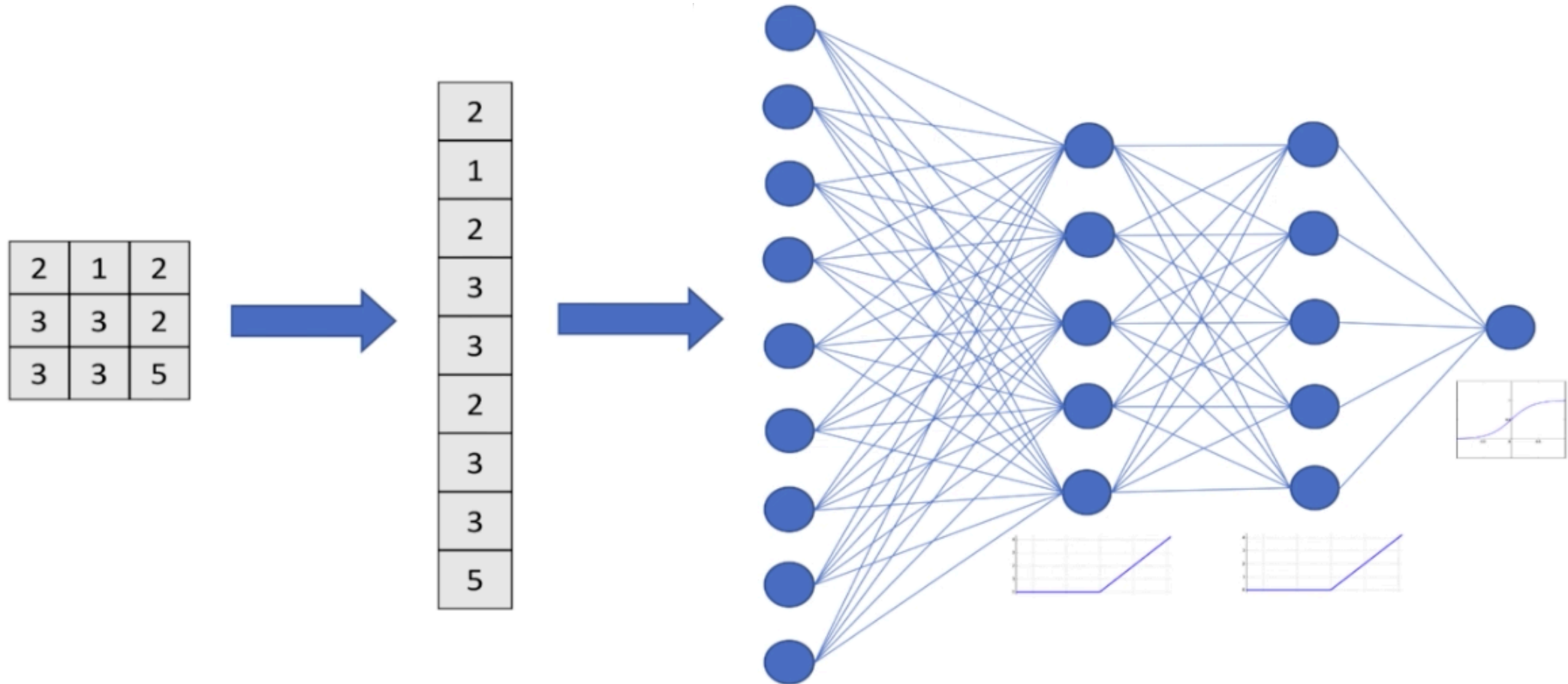
Rede neural convolucional (polling)

As diferentes arquiteturas repetem e combinam essas e outras funções de transformações do sinal.

O resultado da repetição dessas camadas é um conjunto de *features* especializado na tarefa na qual ela foi treinada.



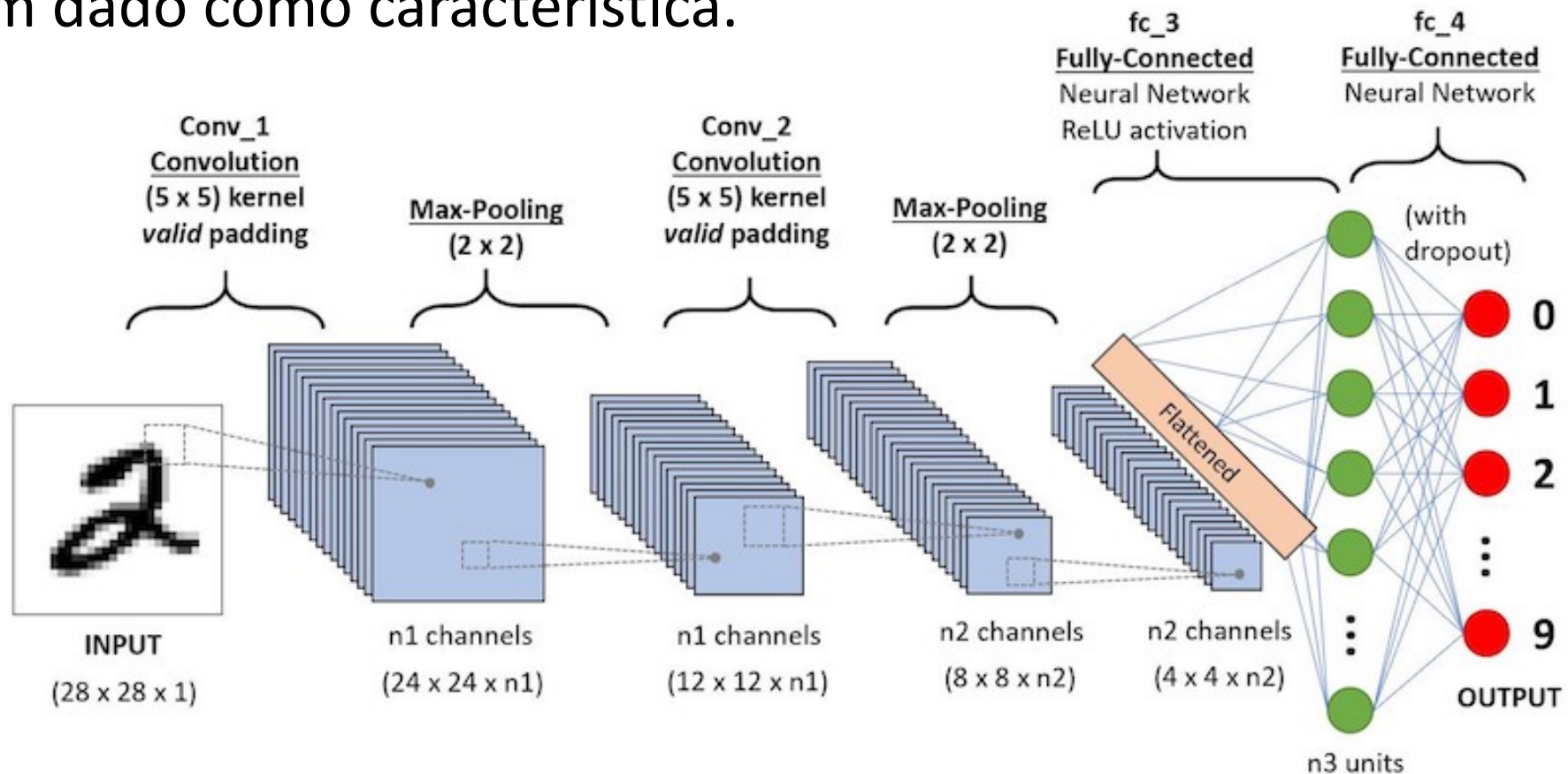
Flattening



Flattening -> Transformar uma matriz para um vetor

Flattening

- Antes de submeter a rede neural, uma etapa adicional se faz necessária, o *flattening*, que ajusta a matriz para um vetor para que cada neurônio da rede recebe um dado como característica.

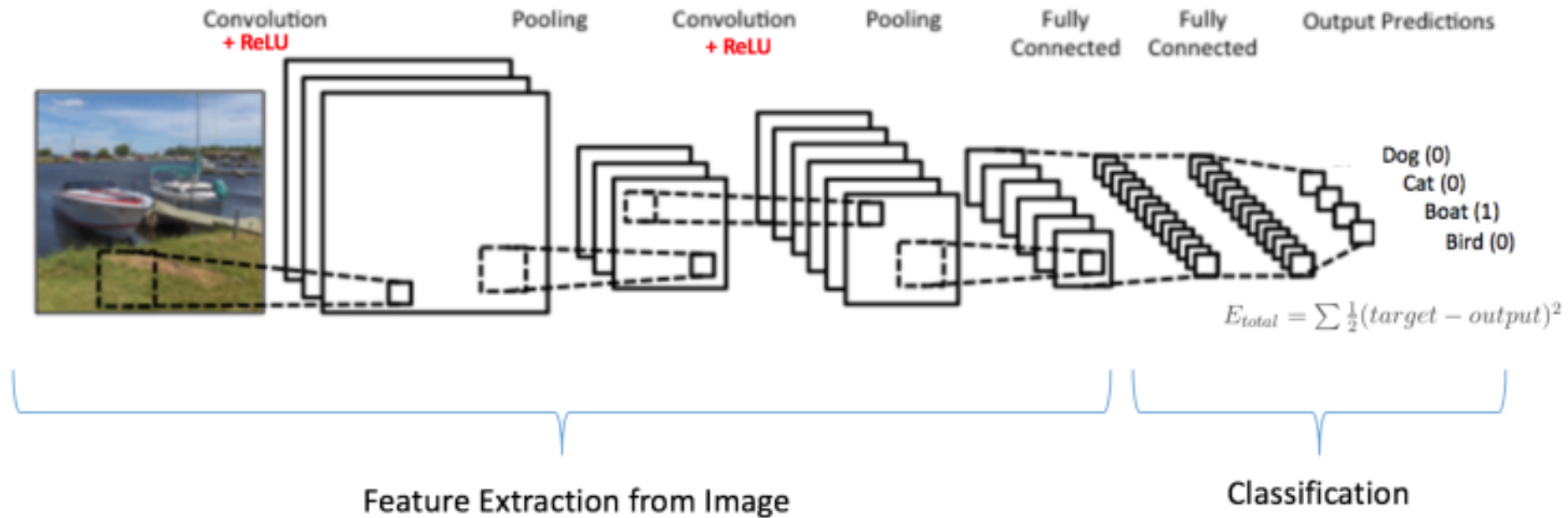


Redes neurais convolucionais (CNN)

- Isso tudo parte do principio de extrair de uma imagens suas características, que são seus pixels.
- O que difere um mago qualquer do mestre dos magos?



Redes neurais convolucionais (CNN)



Atividade

- Criar um banco de imagens (preferencia imagens pequenas) de no mínimo 2 personagens.
- Criar um *dataset* com os valores de características das imagens junto com sua classificação. Exemplo:

caracteristicas	classe
1,0,0,1,0,1,1,1,0,1,0,1,1,0,1,0,0,1,1,1	batman
0,0,0,1,0,1,1,1,0,1,0,1,1,0,1,0,0,1,1,0	superman
1,0,0,1,0,1,1,1,0,1,0,1,1,0,1,0,0,1,1,0	flash
1,0,0,1,0,1,1,1,0,1,0,1,1,0,1,0,0,1,1,1	batman
0,0,0,1,0,1,1,1,0,1,0,1,1,0,1,0,0,1,1,1	aquaman

- Utilizar imagens em tons de cinza.
- Mínimo de 100 imagens.