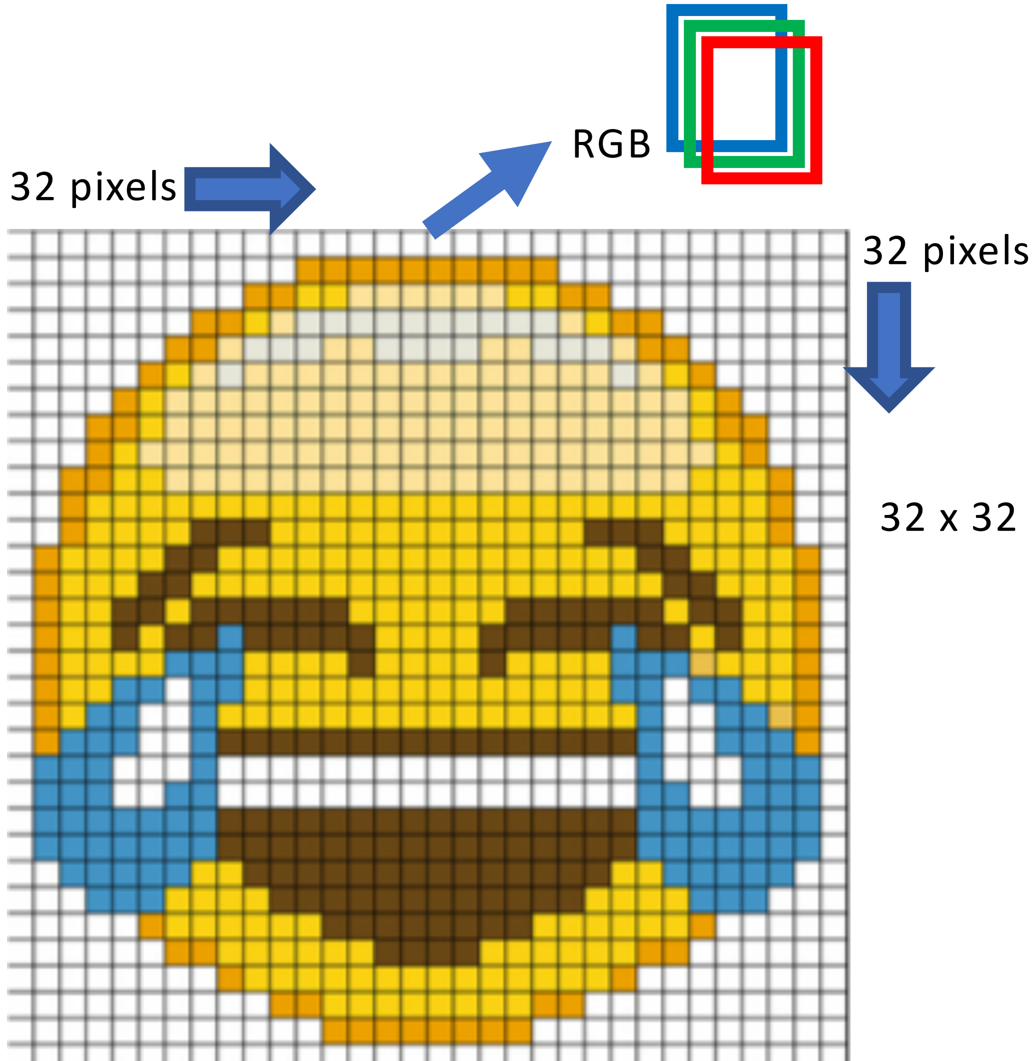
Redes neurais convolucionais

Jones Granatyr

Redes neurais convolucionais (CNN)

- Usado para visão computacional
- Carros autônomos, detecção de pedestres (umas das razões por deep learning funcionar bem)
- Em geral, melhor do que SVM (support vector machines)

Pixels



32 x 32 = 1024 **pixels**

Laranja camisa

Azul calção

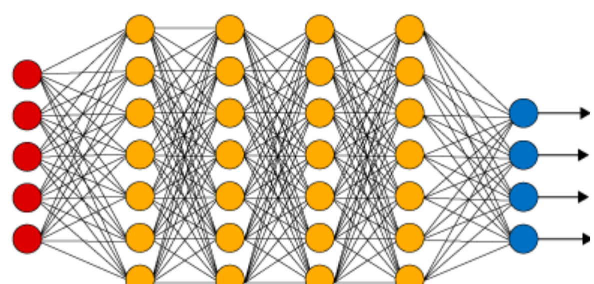
Azul sapato

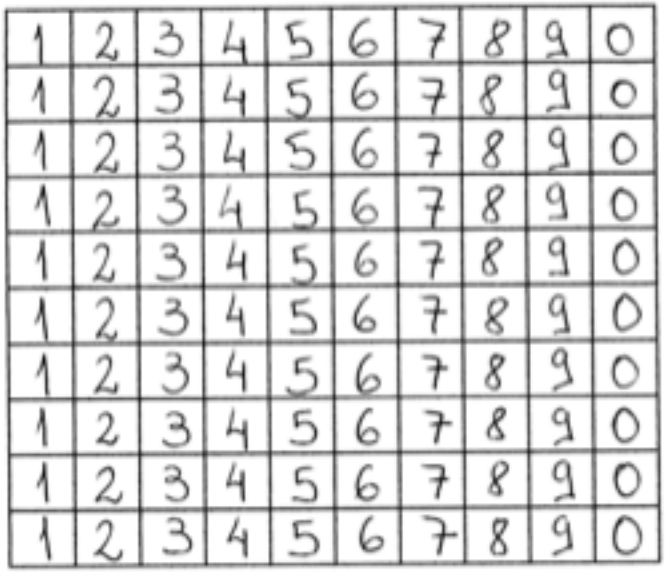
Marrom boca

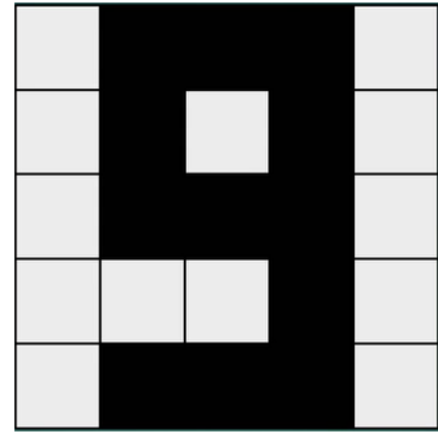
Azul calca

Cinza sapato

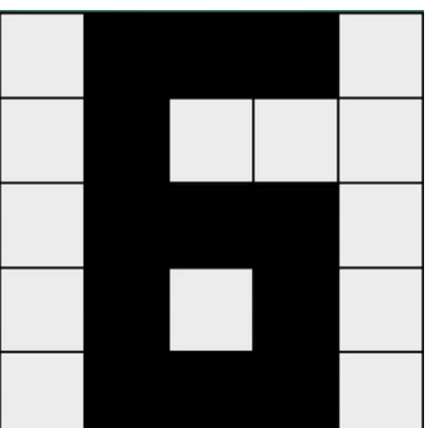
```
8.97,3.45,2.35,0.0,00.00,0.00,Bart
6.75,0.94,0.52,0.00,0.00,0.00,Bart
9.69,4.10,1.56,0.00,0.00,0.00,Bart
0.00,0.00,0.00,4.68,0.66,0.01,Homer
0.00,0.00,0.00,0.12,2.50,0.03,Homer
0.00,0.00,0.00,5.80,0.50,1.28,Homer
```



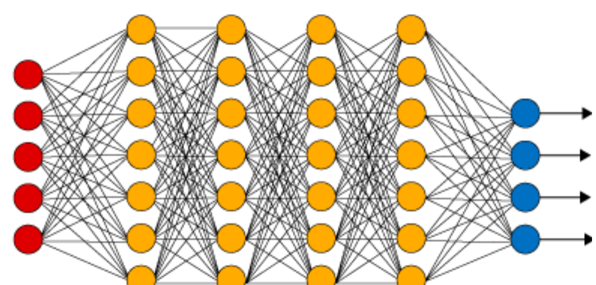




0,1,1,1,0,0,1,0,1,0,0,1,1,1,0,0,0,0,1,0,0,1,1,1,0,**9**

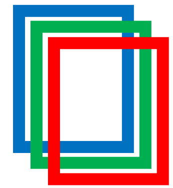


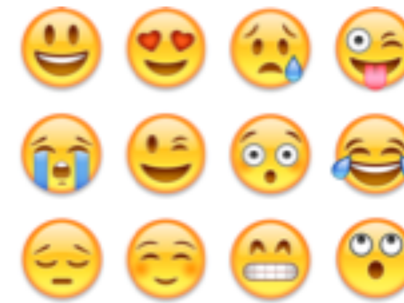
0,1,1,1,0,0,1,0,0,0,0,1,1,1,0,0,1,0,1,0,0,1,1,1,0,**6**

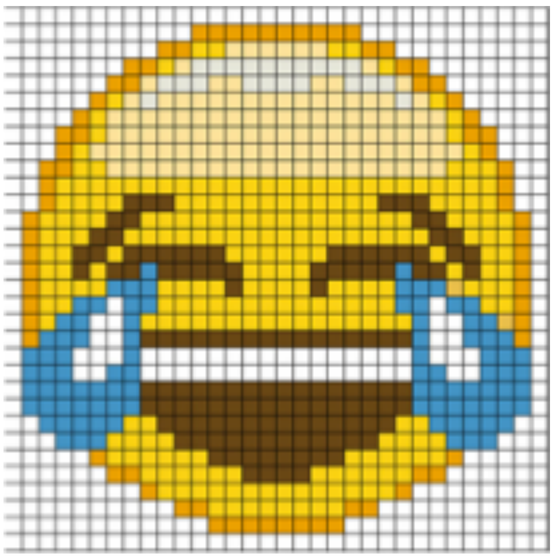


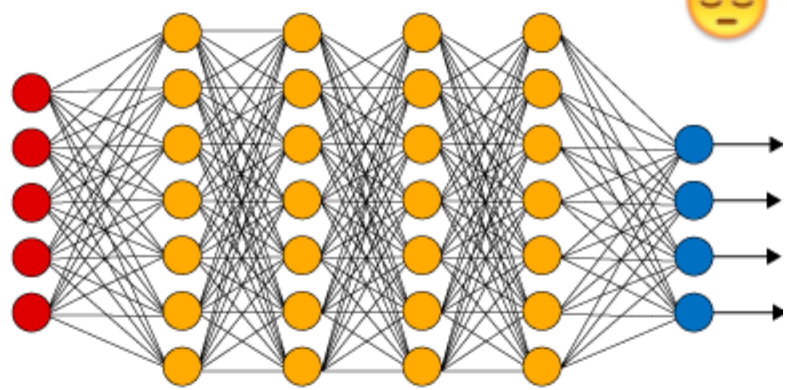
Redes neurais densas x convolucionais

32 x 32 = 1.024 x 3 = **3.072 entradas**









- Não usa todas as entradas (pixels)
- Usa uma rede neural tradicional, mas no começo transforma os dados na camada de entrada
- Quais são as características mais importantes?

Redes neurais convolucionais (CNN)

- Quais características utilizar?
- Para faces
  - Localização do nariz
  - Distância entre os olhos
  - Localização da boca
- Como diferenciar uma face humana de um animal?
- CNN descobre as características

Redes neurais convolucionais (CNN)

- Etapa 1 – Operador de convolução
- Etapa 2 – Pooling
- Etapa 3 – Flattening
- Etapa 4 – Rede neural densa

Etapa 1 – Operador de convolução

- Convolução é o processo de adicionar cada elemento da imagem para seus vizinhos, ponderado por um kernel
- A imagem é uma matriz e o kernel é outra matriz

$$
\begin{aligned}
(f * g)[n] &= \sum_{m=-\infty}^{\infty} f[m]g[n-m] \\
&= \sum_{m=-\infty}^{\infty} f[n-m]g[m].
\end{aligned}
$$

Fonte: https://en.wikipedia.org/wiki/Convolution

Etapa 1 – Operador de convolução

- Explicações sobre os kernels
    - https://en.wikipedia.org/wiki/Kernel_(image_processing)
- Exemplo on-line
    - http://setosa.io/ev/image-kernels/

Etapa 1 – Operador 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

X

1	0	0
1	0	1
0	1	1

Detector de características
(feature detector)

=

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

Etapa 1 – Operador 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

X

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

Etapa 1 – Operador 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

X

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

Etapa 1 – Operador 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

X

1	0	0
1	0	1
0	1	1

Detector de características
(feature detector)

=

0	1	0	1	

Mapa de características
(feature map)

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

Etapa 1 – Operador 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

X

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)

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

Etapa 1 – Operador de convolução

- Com o mapa de características (filter map) a imagem fica menor para facilitar o processamento
- Alguma informação sobre a imagem pode ser perdida, porém o propósito é detectar as partes principais (quanto maior os números melhor)
- O mapa de características preserva as características principais da imagem (olho, boca, nariz, por exemplo)

Etapa 1 – Operador de convolução (Relu)

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

1	0	0
1	0	1
0	1	1

Detector de características
(feature detector)

=

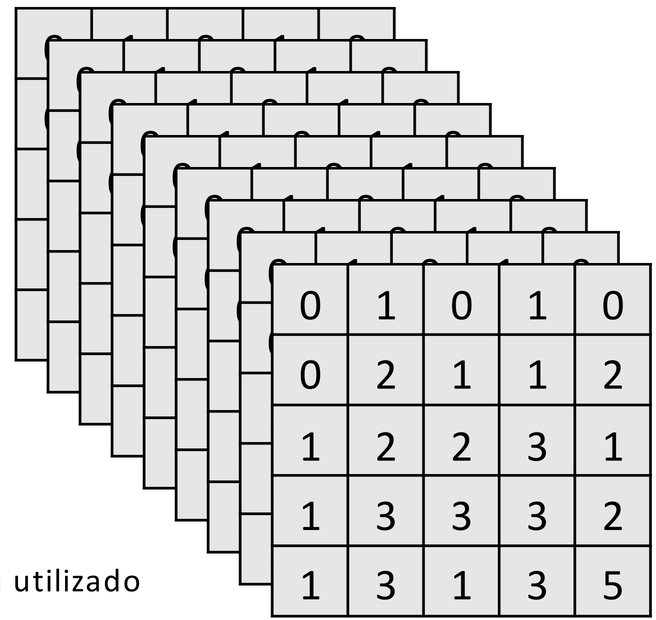
0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

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



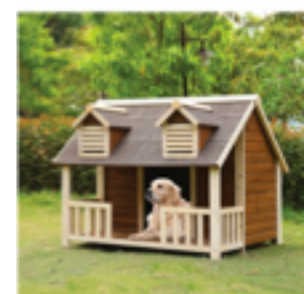
0	1	0	1	0
0	2	1	1	2
1	2	2	3	1
1	3	3	3	2
1	3	1	3	5

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

Etapa 2 - Pooling



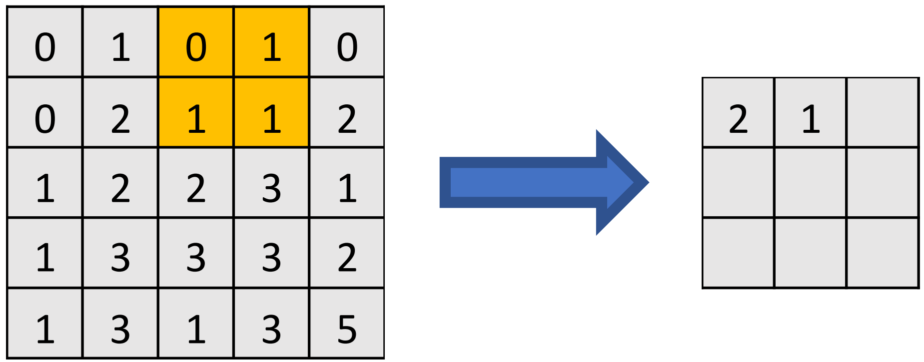
Etapa 2 – Pooling

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		

Etapa 2 – Pooling



Mapa de características
(feature map)

Etapa 2 – Pooling

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

Etapa 2 – Pooling

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		

Etapa 2 – Pooling

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	

Etapa 2 – Pooling

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

Etapa 2 – Pooling

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		

Etapa 2 – Pooling

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	

Etapa 2 – Pooling

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

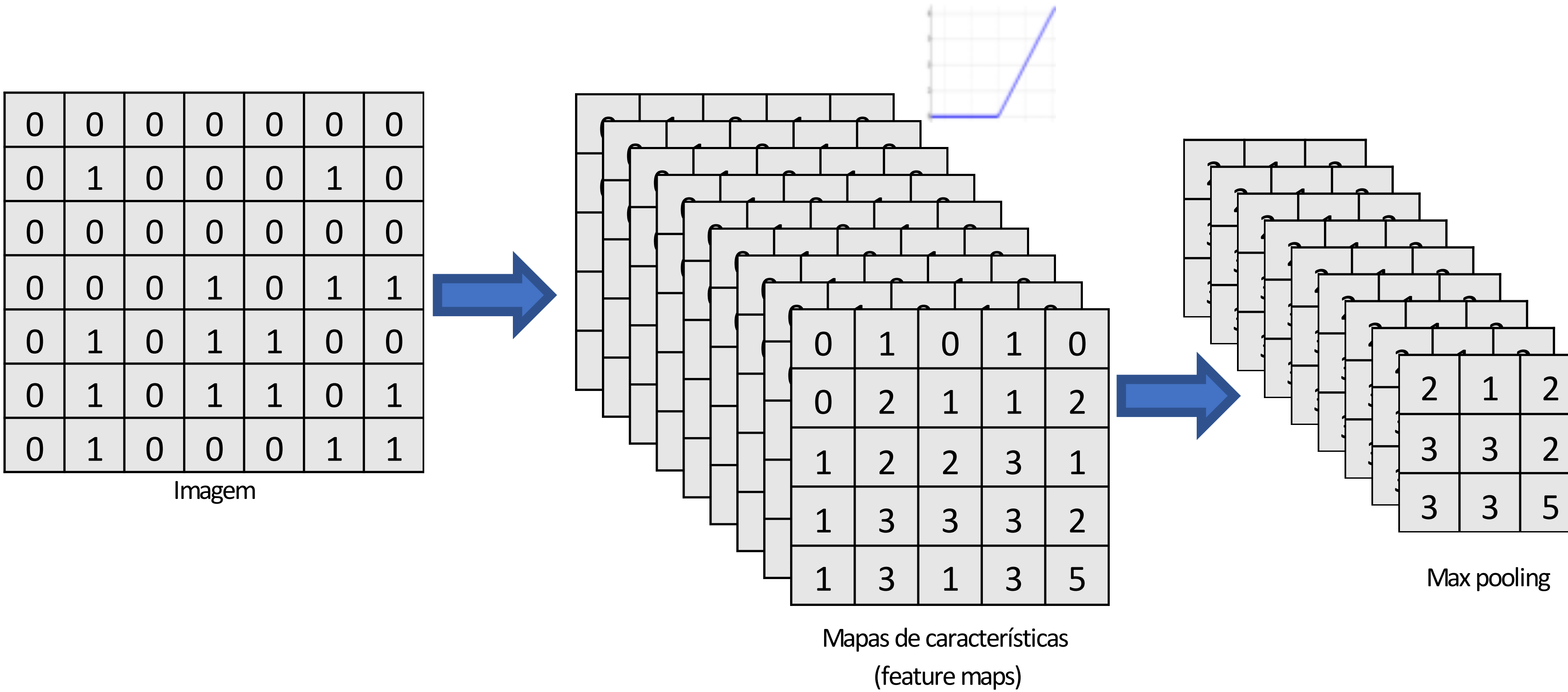
Mapa de características
(feature map)

2	1	2
3	3	2
3	3	5

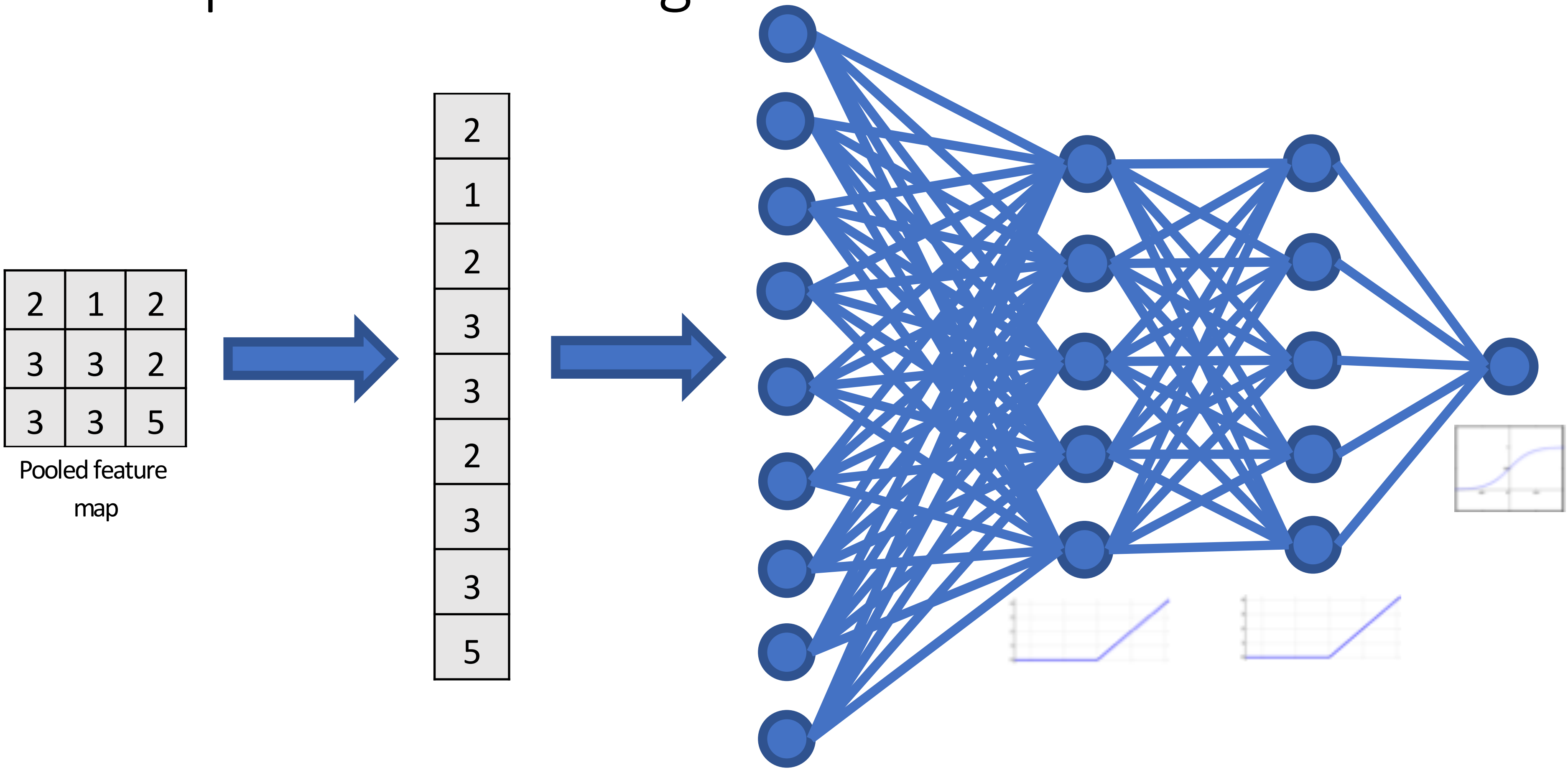
Etapa 2 –Pooling

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

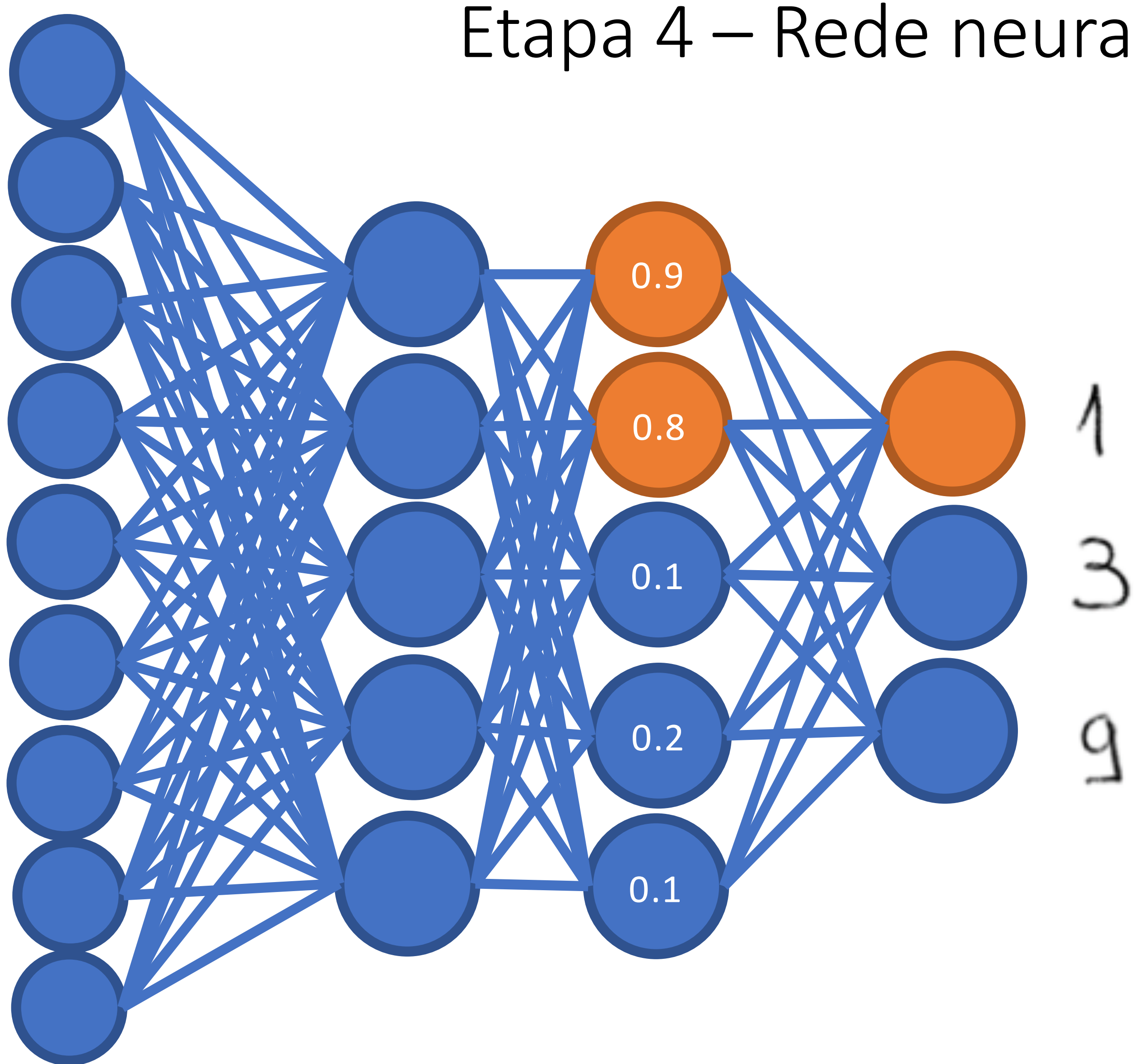
Rede neural convolucional (polling)



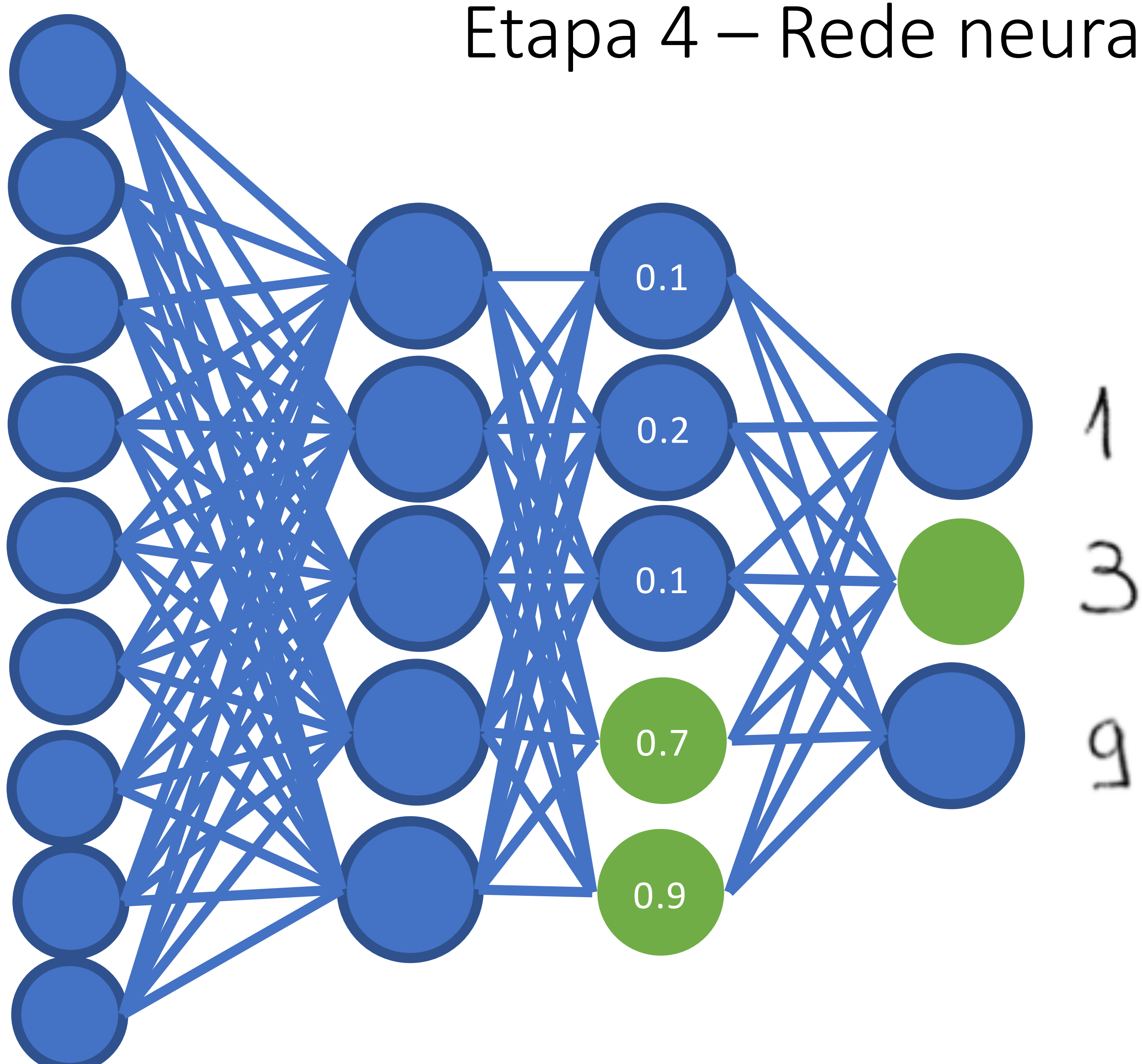
Etapa 3 – Flattening



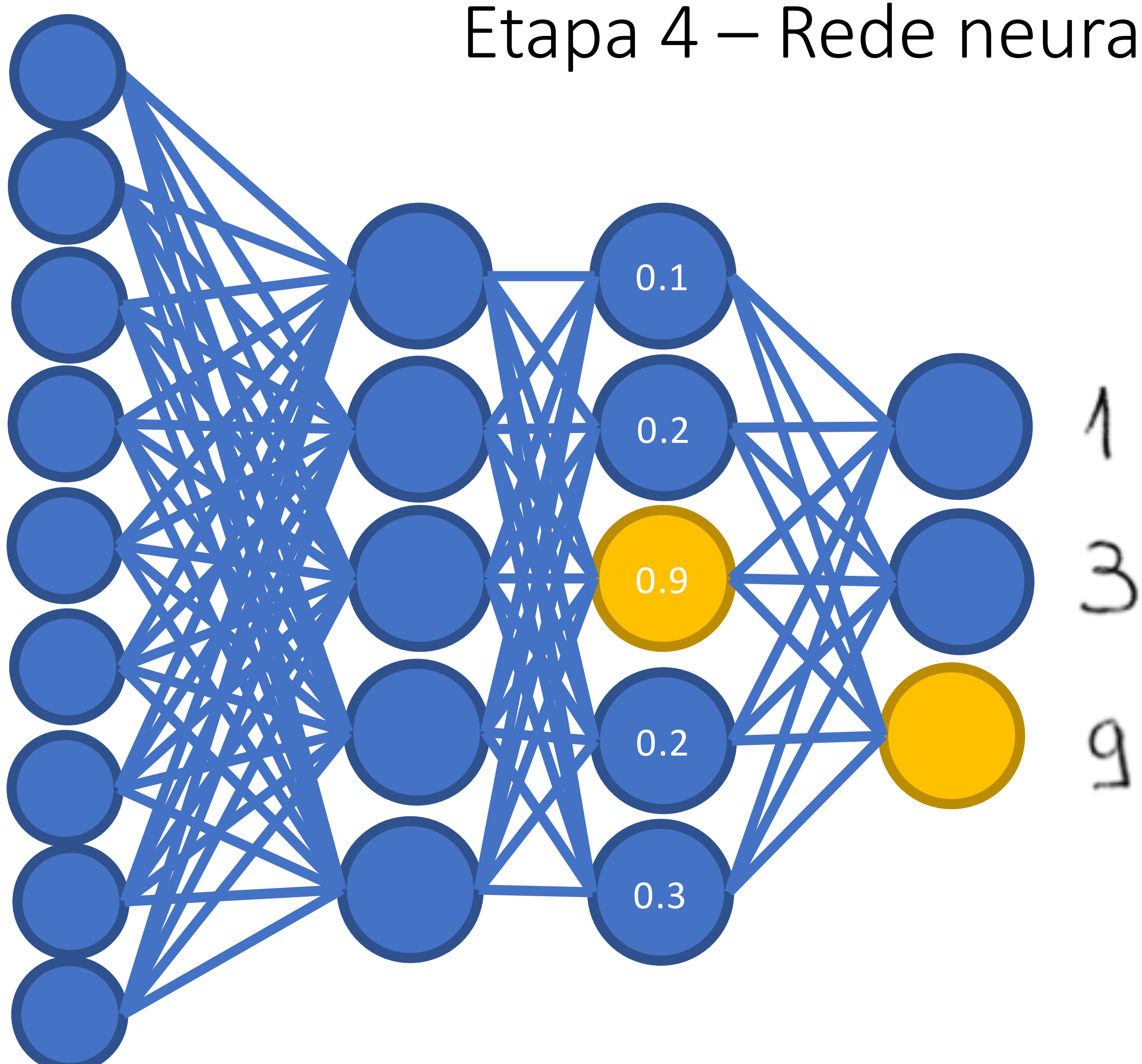
Etapa 4 – Rede neural densa



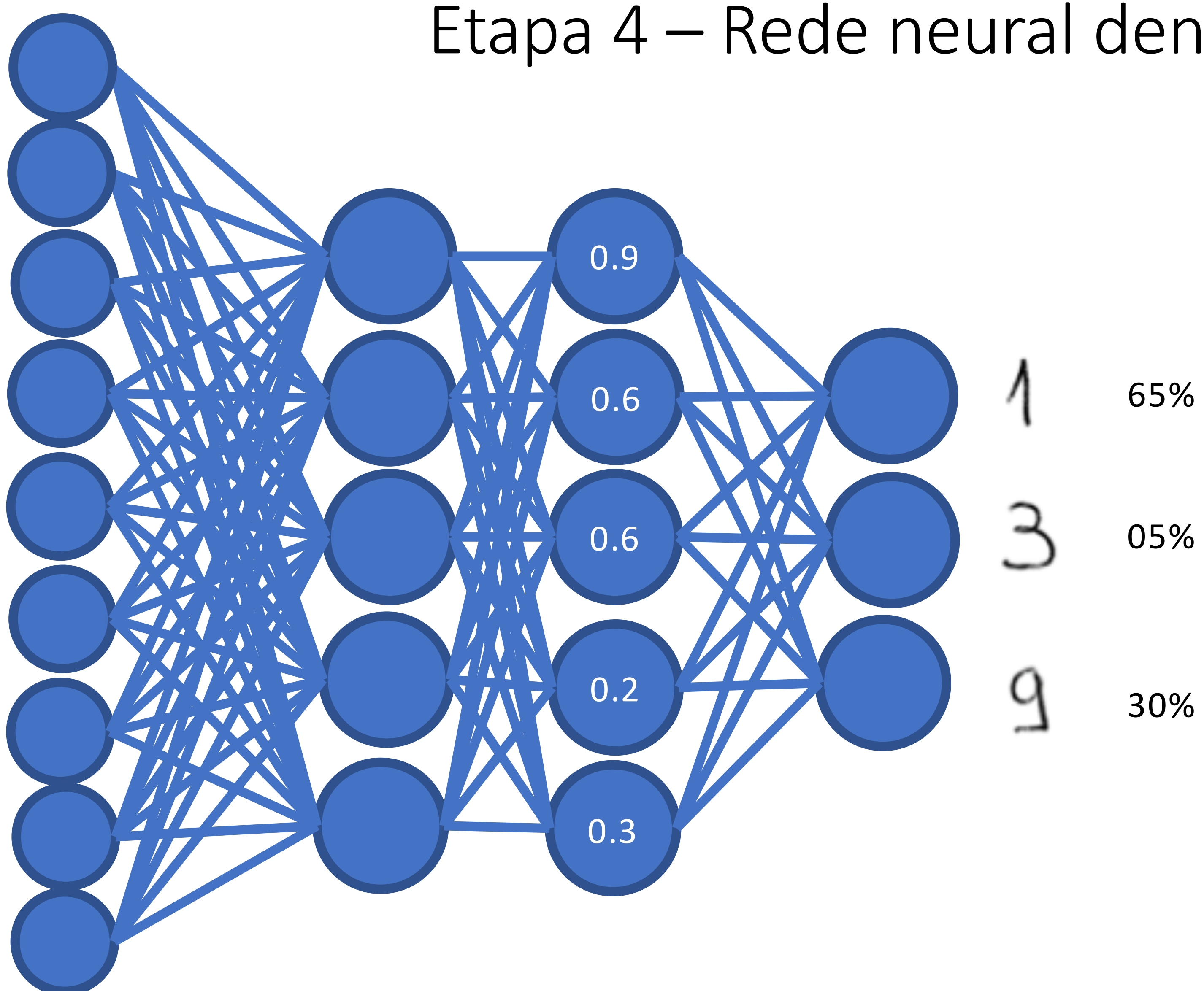
Etapa 4 – Rede neural densa



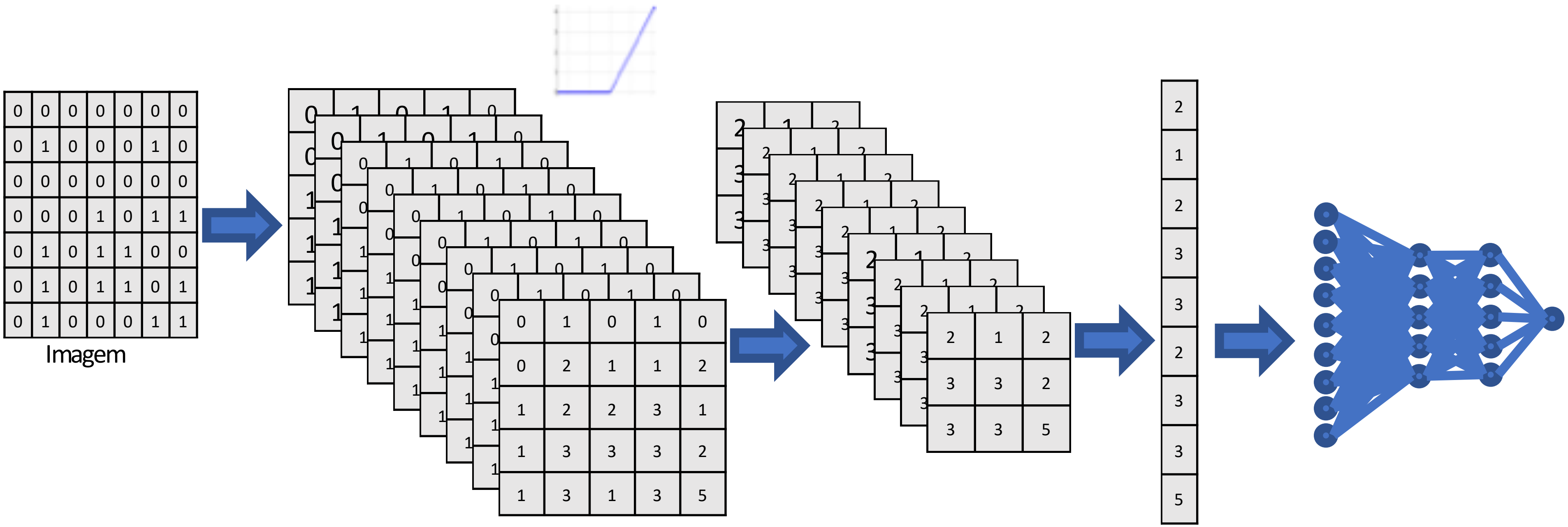
Etapa 4 – Rede neural densa



Etapa 4 – Rede neural densa



Rede neural convolucional



Treinamento com a descida do gradiente

Além do ajuste dos pesos, é feito também a mudança do detector de características

Conclusão