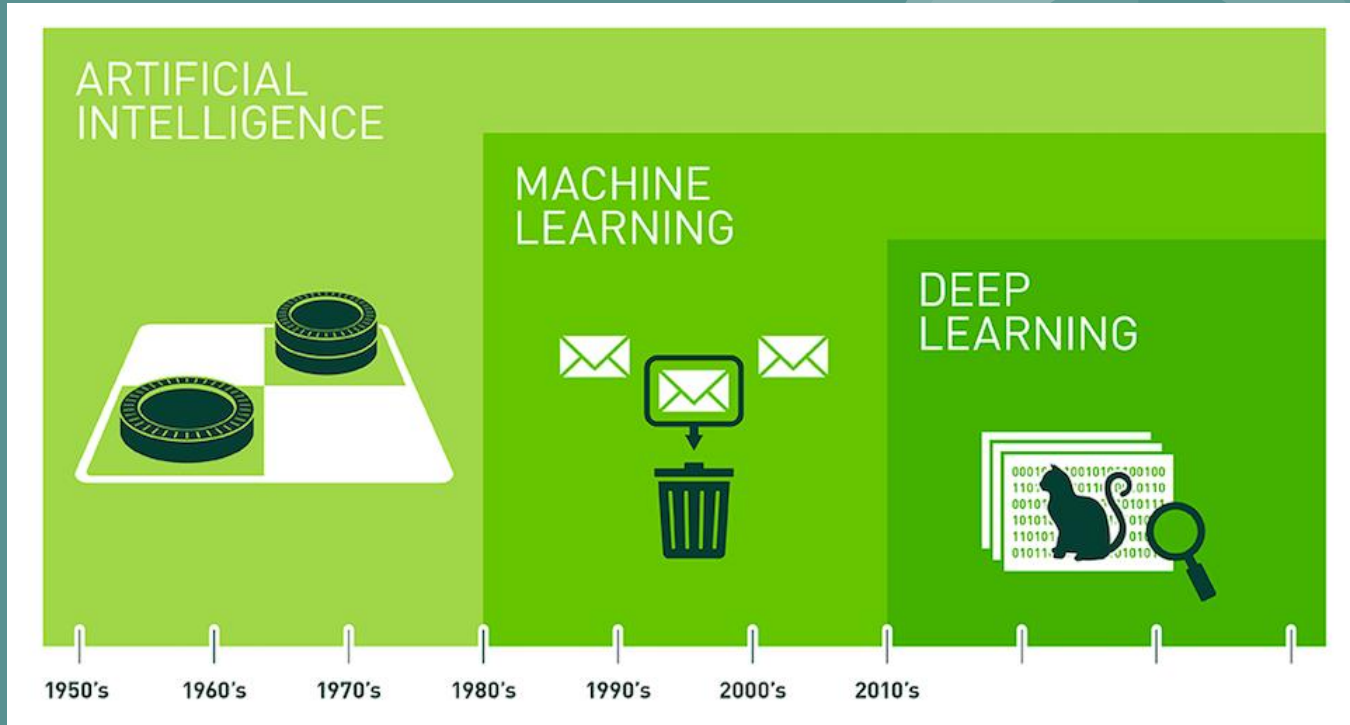


# Redes Neurais Convolucionais

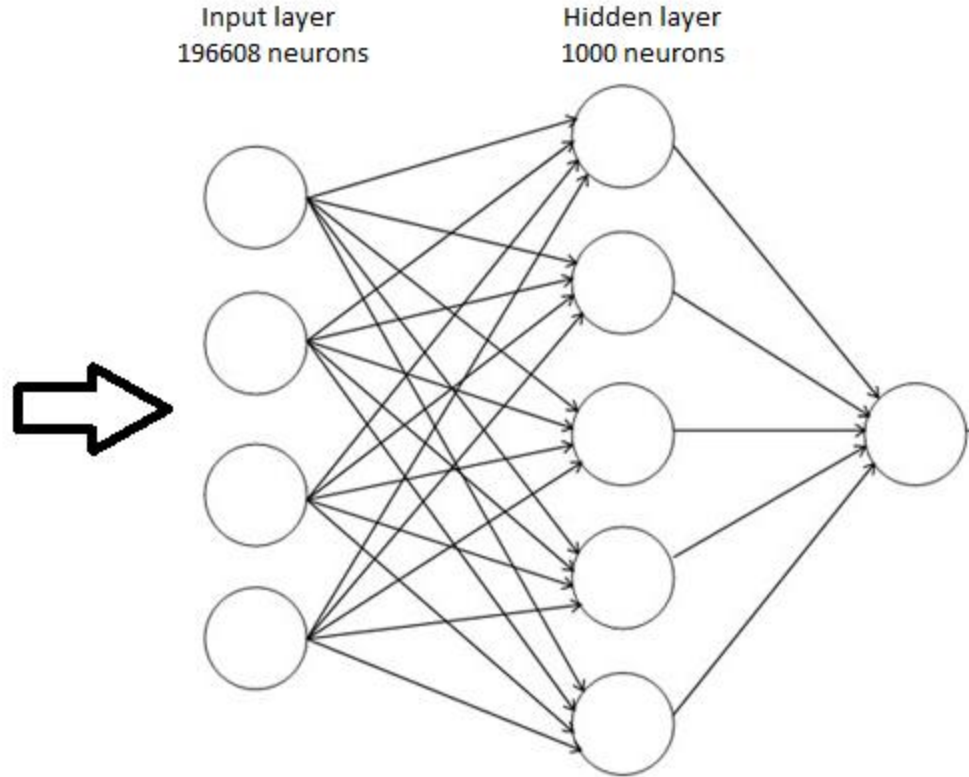
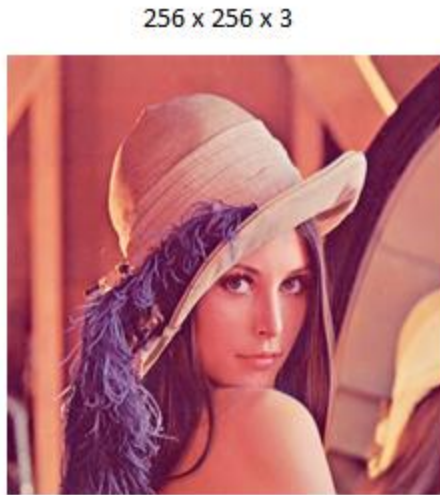
Hialo Muniz Carvalho



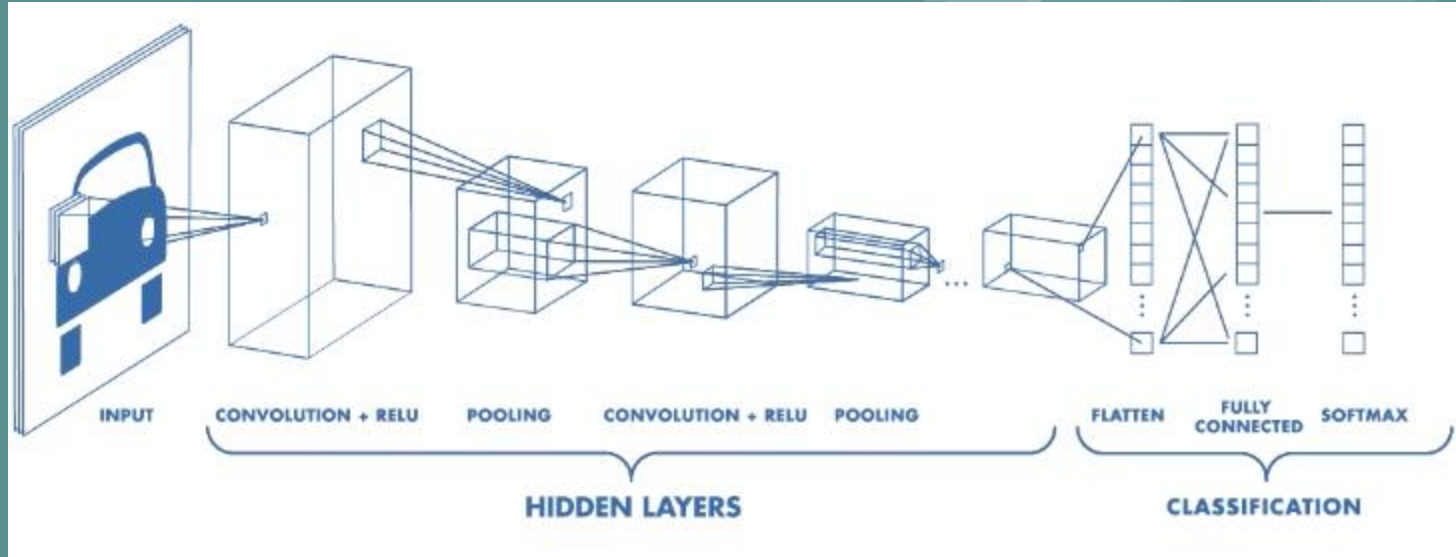
# IA x Machine Learning x Deep Learning



## Rede Neural FC (Fully-Connected)



# Estrutura de uma Rede Neural Convolucional



# Estrutura de uma Rede Neural Convolucional

- Camada de convolução
  - *Filtros*
  - *Strides (passos)*
  - *Padding*
- Camada de não-linearidade (ReLU)
- Camada de *pooling*
- Camada Fully-Connected (FC)



# Aplicações de uma Rede Neural Convolutacional

- Processamento de imagens
- Reconhecimento facial
- Detecção de objetos
- Processamento de documentos
- Natural Language Processing (NLP)



# Camada de convolução

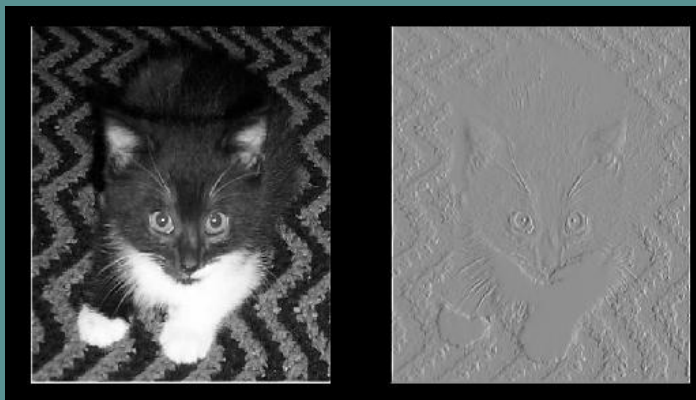
- Entrada: matriz representando a imagem e um filtro (também conhecido como *kernel*)
- Saída: *mapa de características* (*feature map*)
- Extração de características da imagem

1x1	1x0	1x1	0	0
0x0	1x1	1x0	1	0
0x1	0x0	1x1	1	1
0	0	1	1	0
0	1	1	0	0

4		

# Tipos de filtros

- Detecção de bordas
- Detecção de ângulos
- *Blurring*

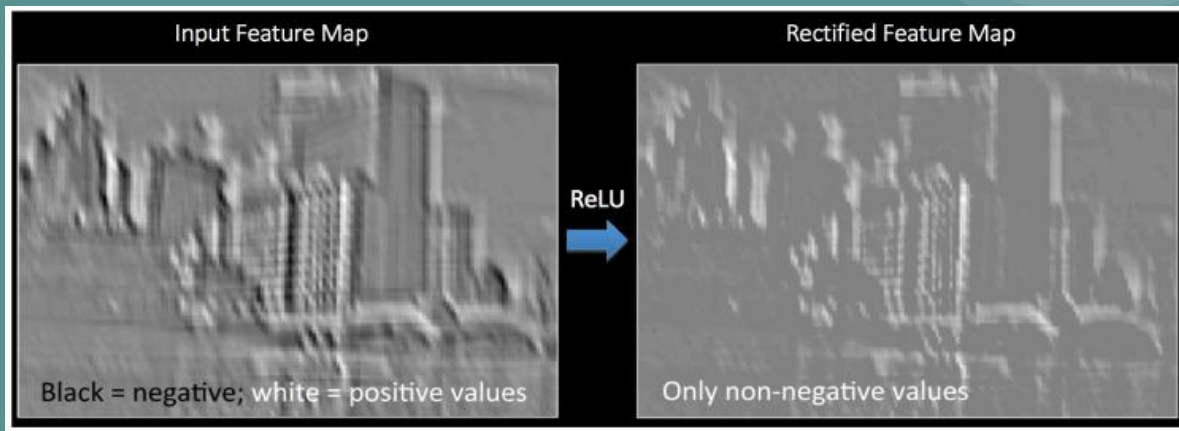
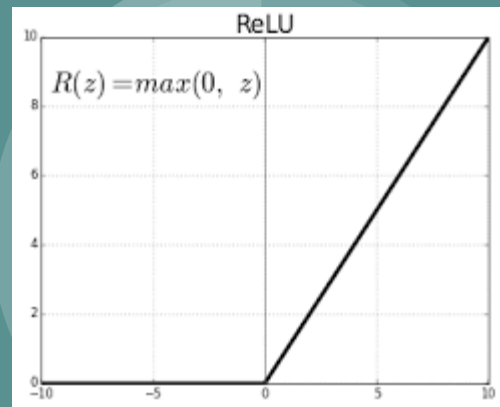


Operation	Filter	Convolved Image
<b>Identity</b>	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
<b>Edge detection</b>	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
<b>Sharpen</b>	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
<b>Box blur</b> (normalized)	$\frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
<b>Gaussian blur</b> (approximation)	$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	



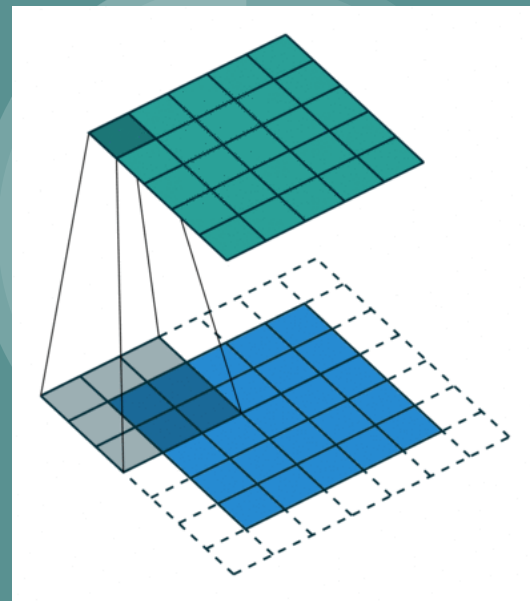
# Tipos de filtros

- Adiciona não-linearidade nas informações obtidas na convolução
- Melhora o tempo de processamento durante o treinamento da rede neural
- *Podemos usar vários tipos de funções não-lineares, como tangente hiperbólica, sigmóide ou ReLU (Rectified Linear Unit)*



# Padding e strides

- *Padding*: Preenchimento da imagem para que o filtro seja capaz de processar a imagem como um todo
- Tipos de *padding*:
  - *Valid*: reduz o tamanho total da imagem, sem se preocupar com o tamanho do filtro
  - *Same*: realiza *padding* na entrada para que o mapa de características resultante tenha o tamanho original da entrada
- *Strides* (passos): tamanho do deslizamento do filtro durante a convolução



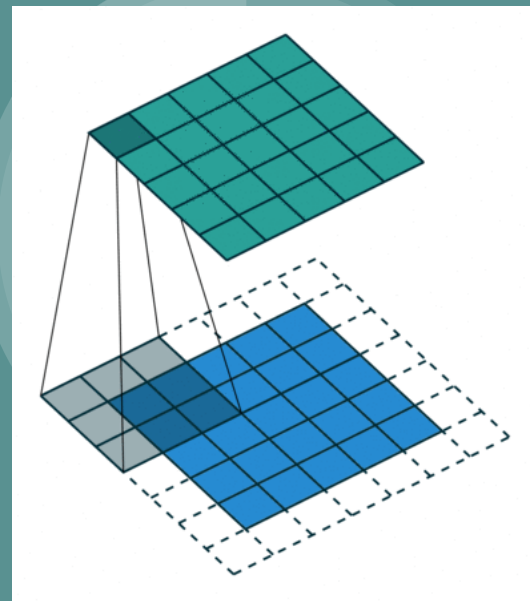
# Padding e strides

- Ao final da convolução, o tamanho do mapa de características resultante é igual a:

$$( (T - F + 2P) / S ) + 1$$

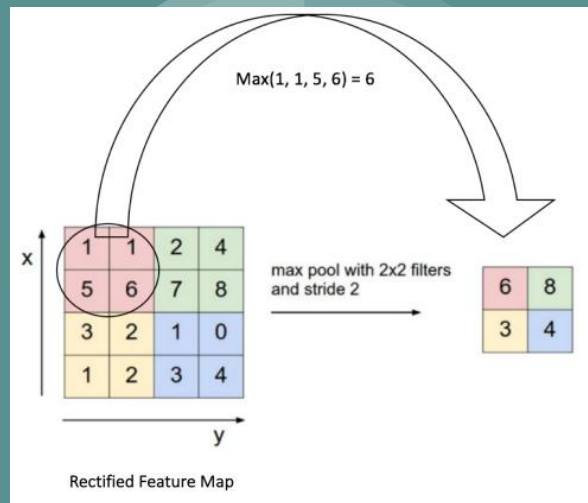
Aonde:

- T é o tamanho da imagem (altura/largura)
- F é o tamanho do filtro
- P é o padding utilizado
- S é o tamanho do passo utilizado



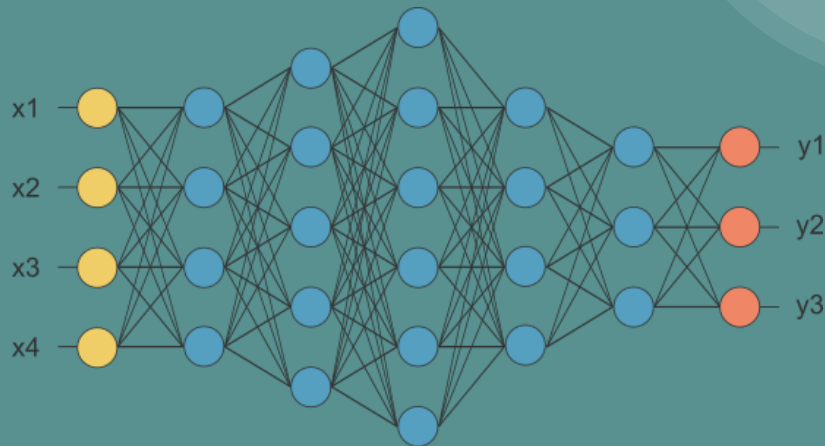
# Camada de *pooling*

- Redução de dimensionalidade
- Controle contra *over-fitting*
- *Tipos de pooling*:
- Métodos comuns de *pooling*:
  - *Max-pooling*
  - *Average pooling*
  - *Sum pooling*

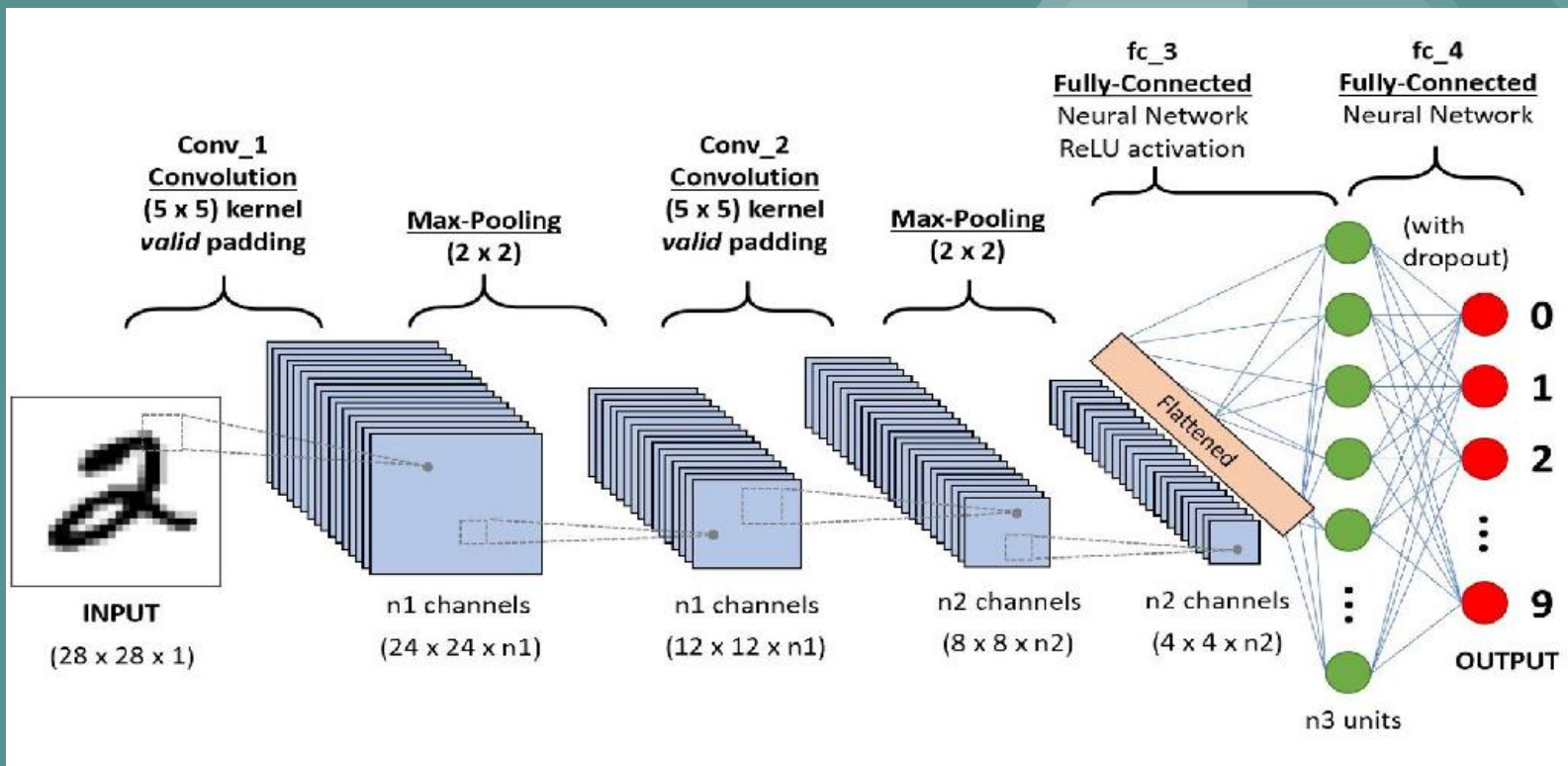


# Camada Fully-Connected (FC)

- Responsável por aprender as características extraídas durante as etapas de *convolução/pooling*
- Semelhante a uma rede neural convencional
- Também conhecida como *flatten layer*



# Exemplo de Rede Neural Convolucional



# Referências

- <https://ahmedbesbes.com/understanding-deep-convolutional-neural-networks-with-a-practical-use-case-in-tensorflow-and-keras.html>
- <https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148>
- <https://medium.com/datadriveninvestor/convolutional-neural-networks-e0d25a3799f6>
- <https://medium.com/@eternalzer0dayx/demystifying-convolutional-neural-networks-ca17bdc75559>
- <https://medium.com/data-science-brigade/a-diferen%C3%A7a-entre-intelig%C3%A7%C3%A2ncia-artificial-machine-learning-e-deep-learning-930b5cc2aa42>
- <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- <http://cs231n.github.io/convolutional-networks/>

# Exercício!

