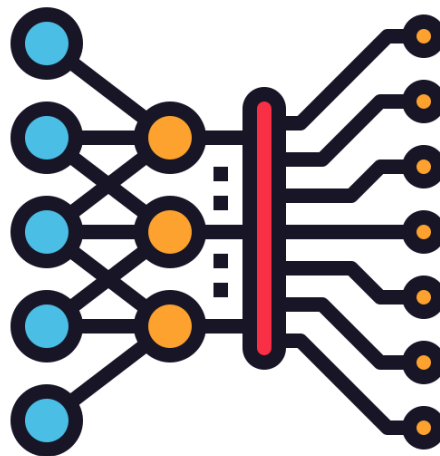


# Propriedades Matemáticas

**Prof. Dr. Diego Bruno**

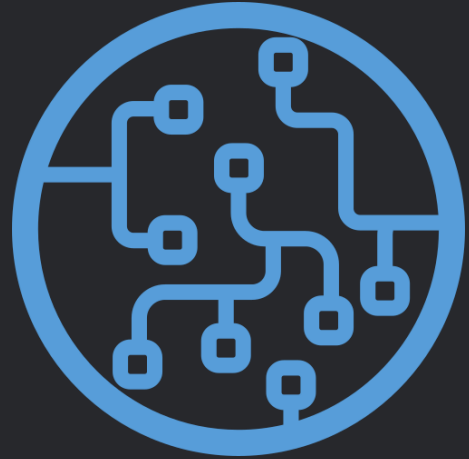
Education Tech Lead na DIO

Doutor em Robótica e *Machine Learning* pelo ICMC-USP



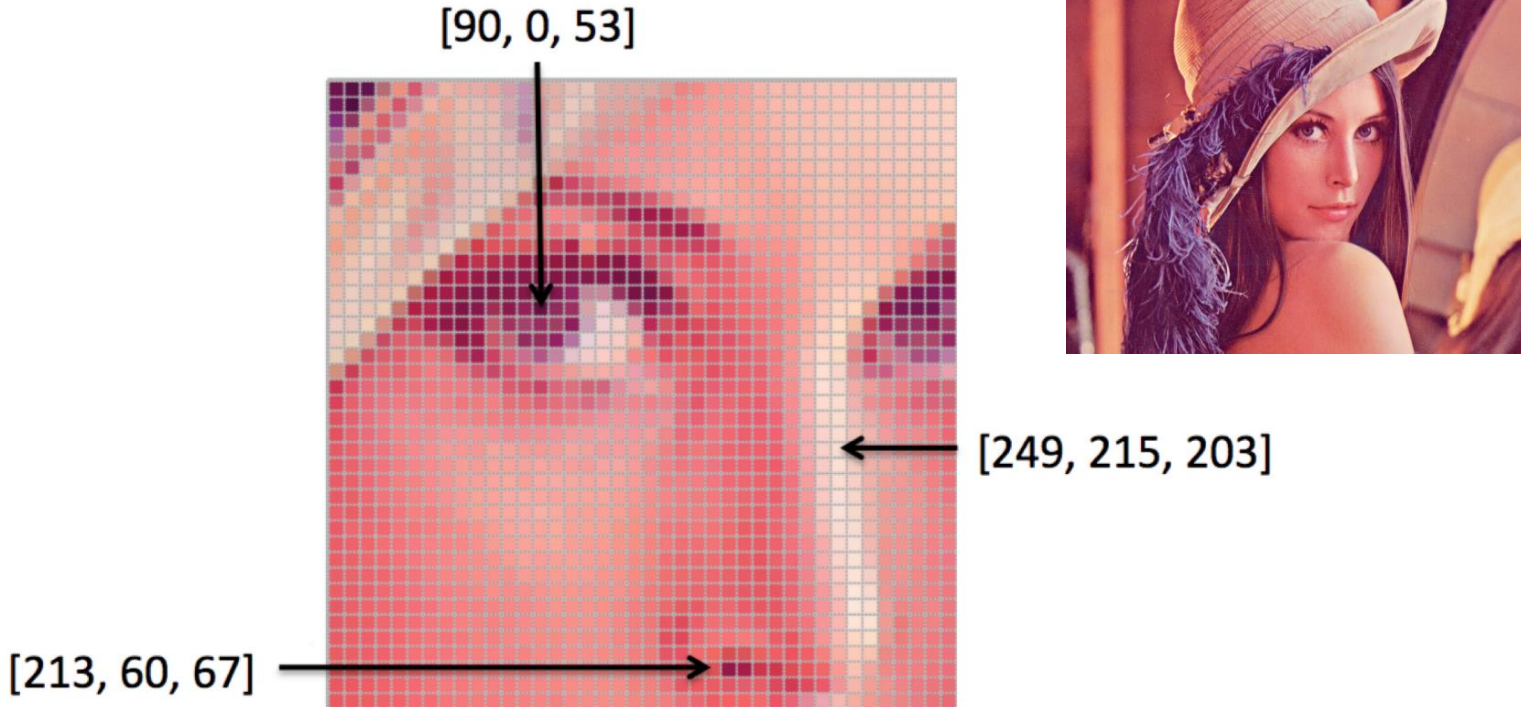
# Algoritmos Convolucionais

*Machine Learning*



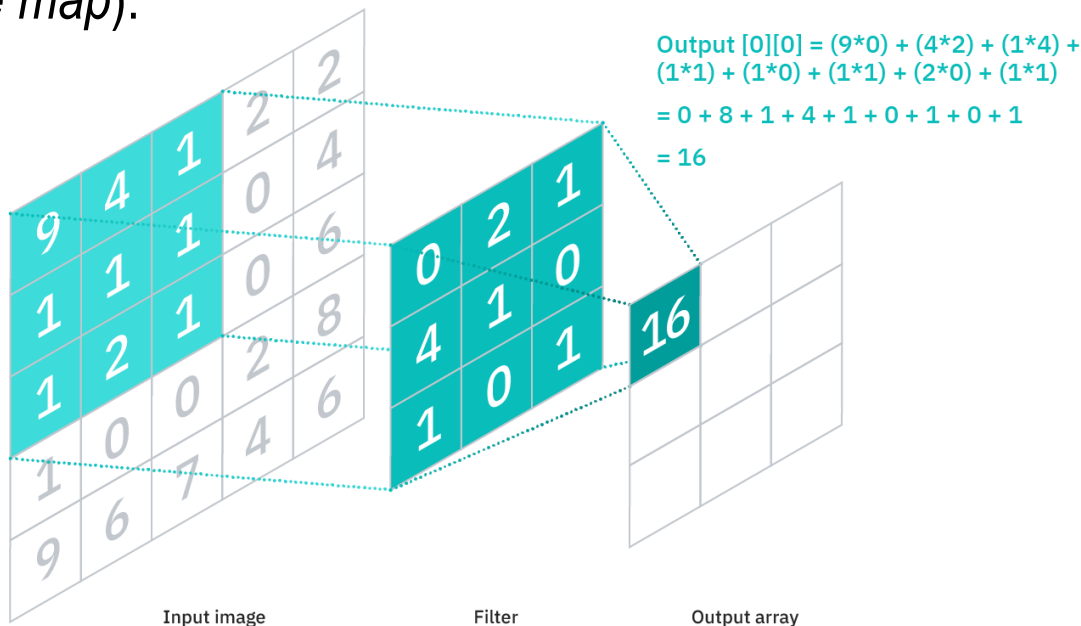
# Propriedades Matemáticas

→ Como uma imagem é representada?



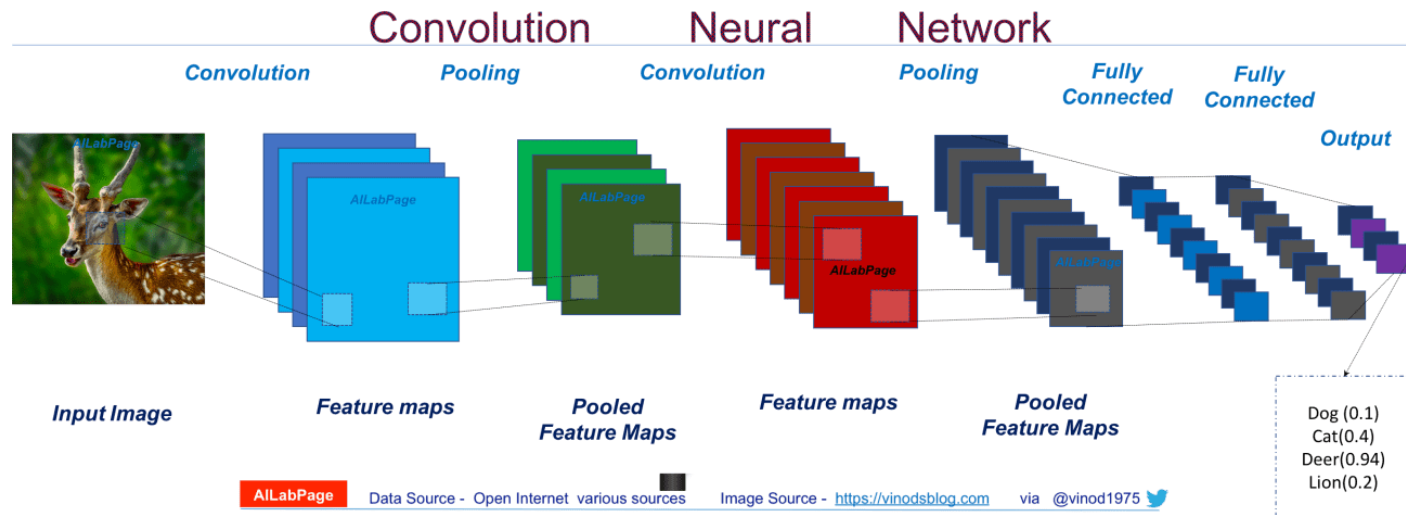
# Propriedades Matemáticas

→ Matematicamente, uma convolução é uma **operação linear** que a **partir de duas funções**, gera uma terceira (normalmente chamada de *feature map*):



# Propriedades Matemáticas

→ No contexto de imagens, podemos entender esse processo como um **filtro/kernel** que transforma uma imagem de entrada.



# Propriedades Matemáticas

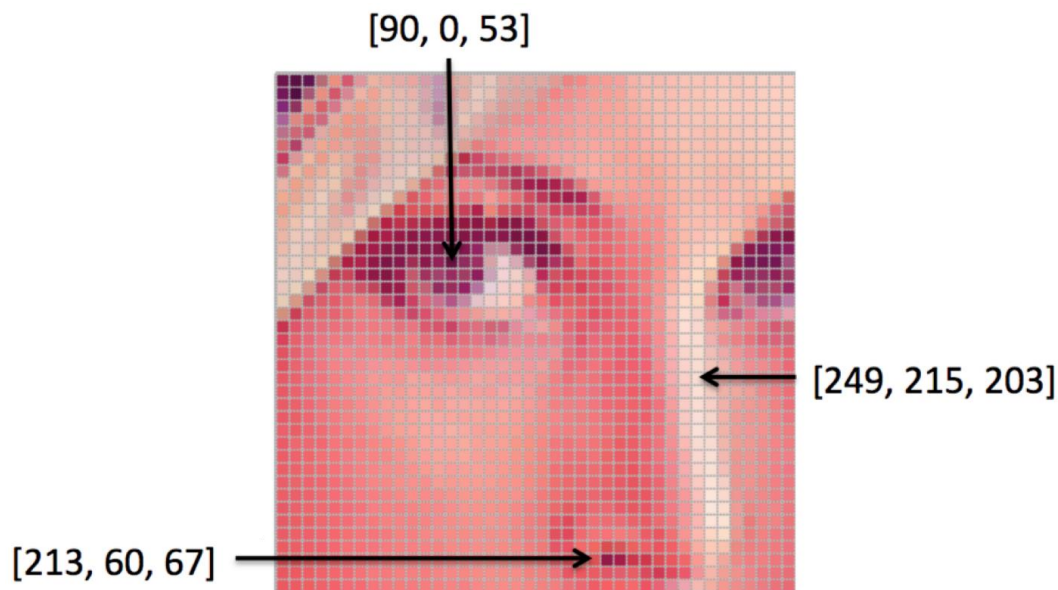
→ Um **kernel** é uma *matrix* utilizada para uma operação de multiplicação de matrizes. Esta operação é aplicada diversas vezes em diferentes regiões da imagem. A cada aplicação, a região é alterada por um parâmetro conhecido como *stride*.

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		

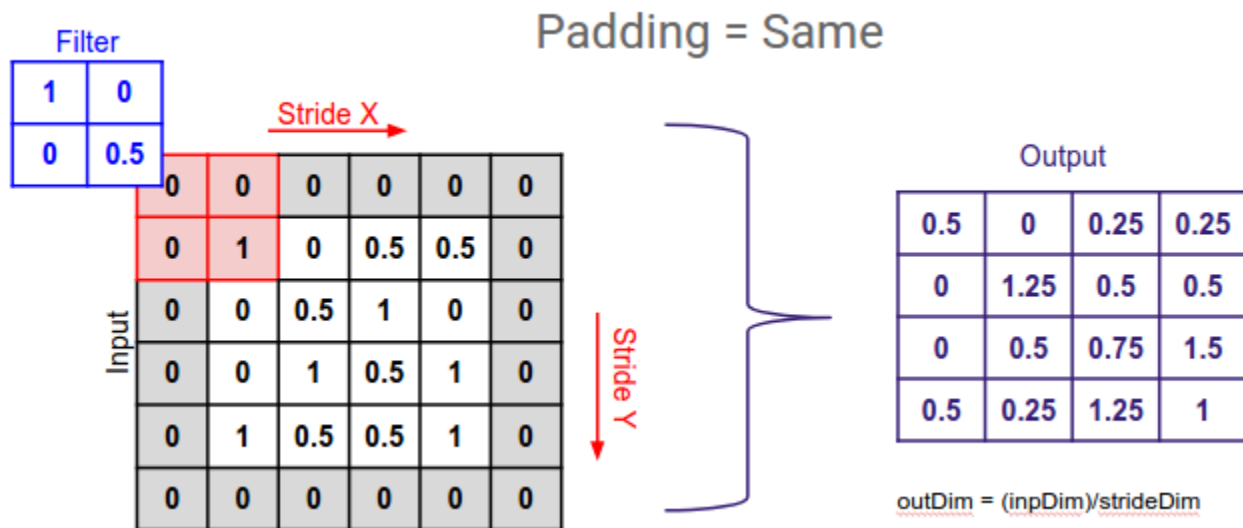
# Propriedades Matemáticas

→ Muitas convoluções podem impactar na assertividade da CNN se o tamanho da **imagem for muito reduzido**. Para contornar esse cenário, normalmente é utilizado o conceito de **Padding**.



# Propriedades Matemáticas

→ **Padding** é um processo em que alguns pixels são adicionados ao redor da imagem antes da operação de convolução, de forma a manter a dimensionalidade na imagem resultante durante a operação.

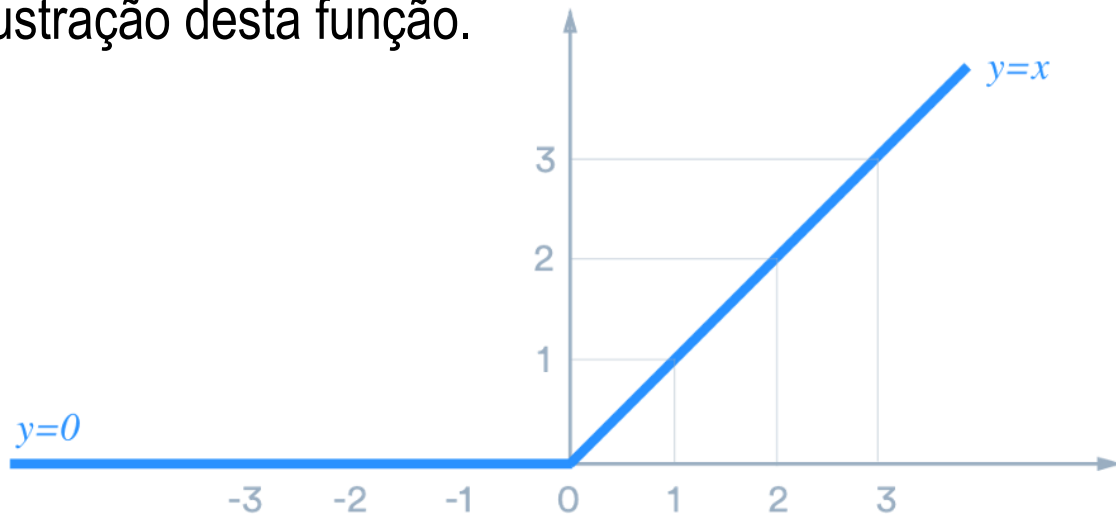




# Propriedades Matemáticas

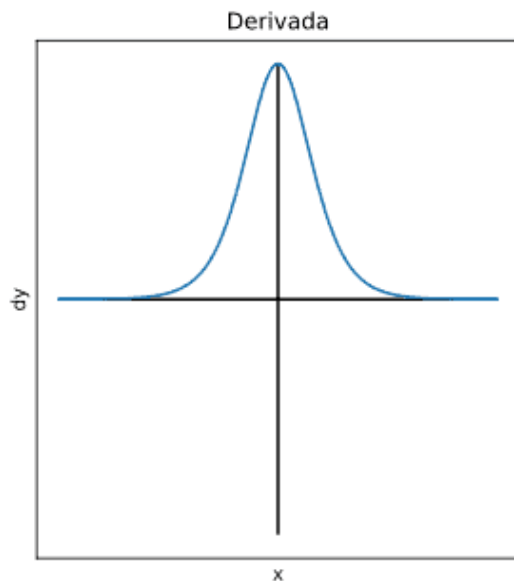
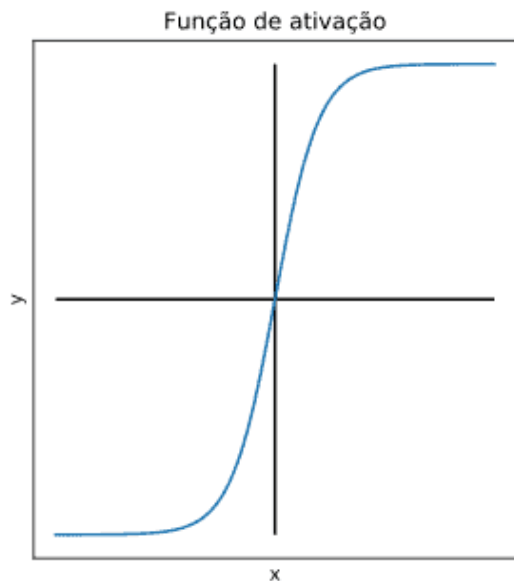
→ Uma rede neural sem função de ativação **torna-se um modelo linear**.  
Se o seu problema é linear, existem outros modelos mais simples...

Matematicamente a função ReLU é definida como  $y = \max(0, x)$ . O gráfico a seguir é a ilustração desta função.



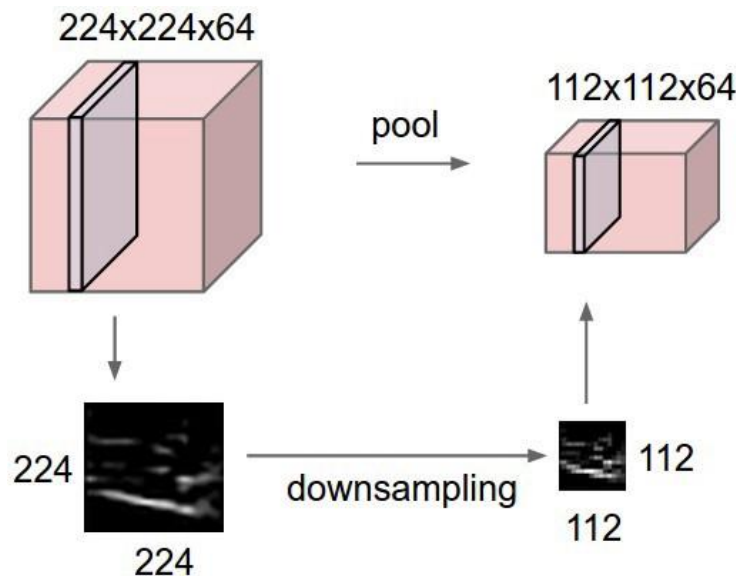
# Propriedades Matemáticas

→ **ReLU** é uma abreviação Unidade Linear Retificada. Ela produz resultados no intervalo  $[0, \infty]$ . A **função ReLU** retorna 0 para todos os valores negativos, e o próprio valor para valores positivos.



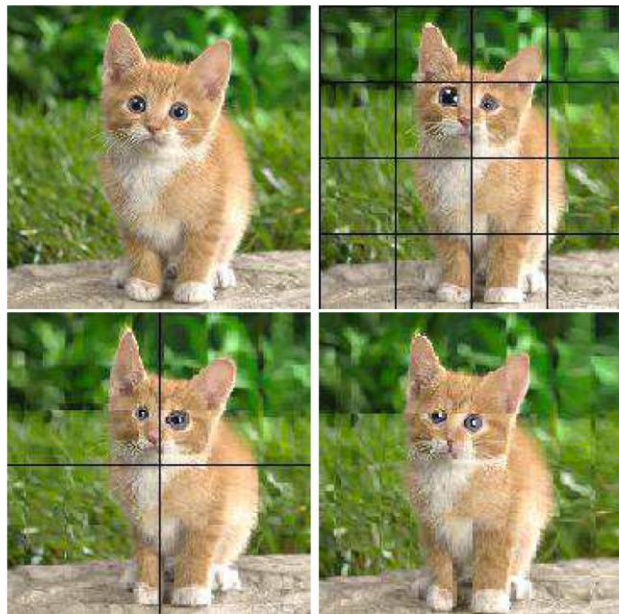
# Propriedades Matemáticas

→ **Pooling** é um processo de **downsampling**. É um processo simples de redução da dimensionalidade/*features maps*. Em uma forma simples de pensar, podemos entender essa transformação como uma redução do tamanho da imagem..

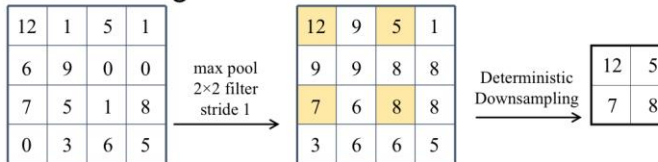


# Propriedades Matemáticas

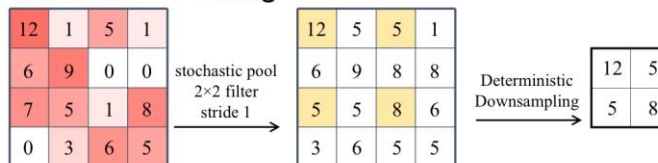
→ A motivação é diminuir sua variância a pequenas alterações e também de reduzir a quantidade de parâmetros treinados pela rede.



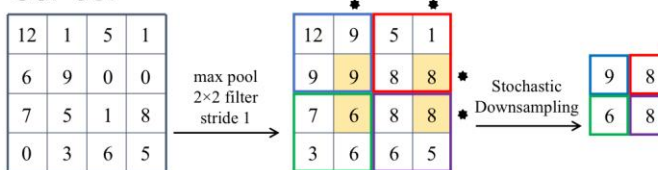
Max Pooling



Stochastic Pooling



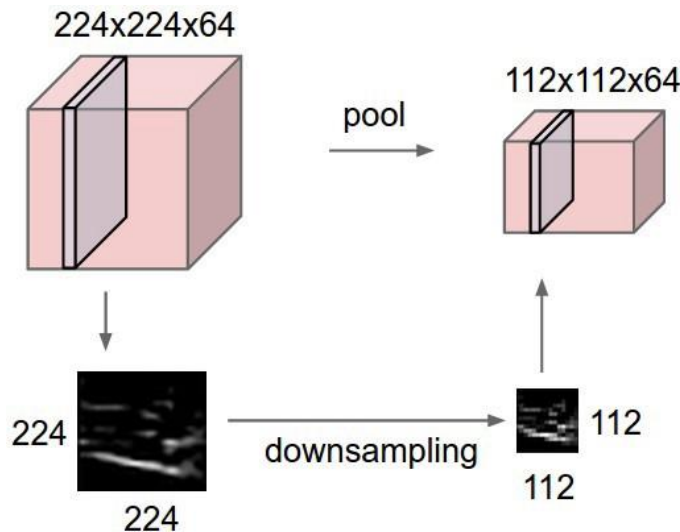
S3Pool



# Propriedades Matemáticas

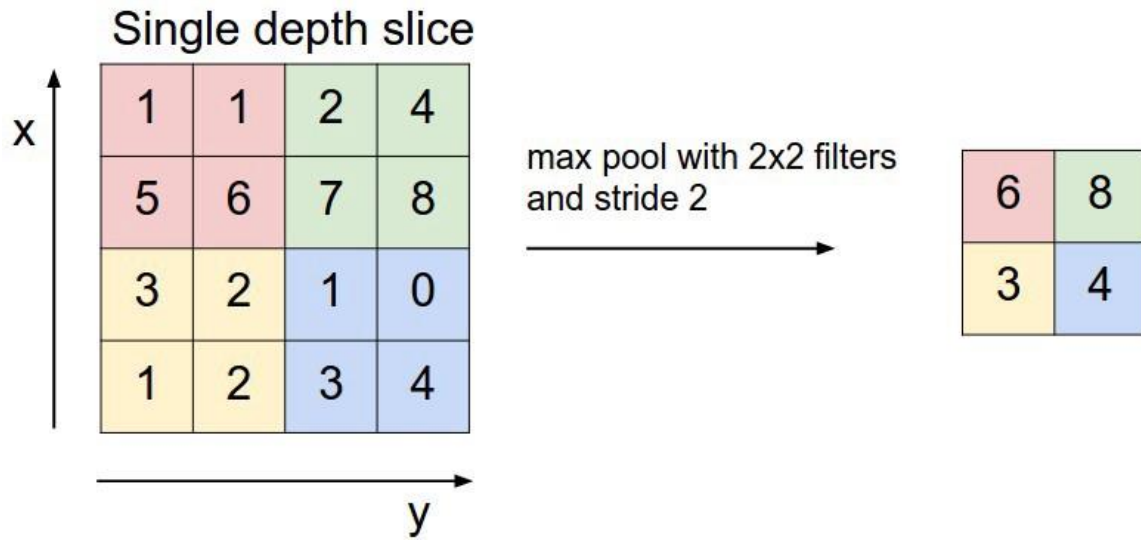
→ Existem 3 operações:

*Pooling* (*MaxPooling*, *SumPooling*, *AveragePooling*). Todas elas seguem o mesmo princípio e só se diferem na forma como calculam o valor final. A mais utilizada nos dias de hoje é a *MaxPooling*.



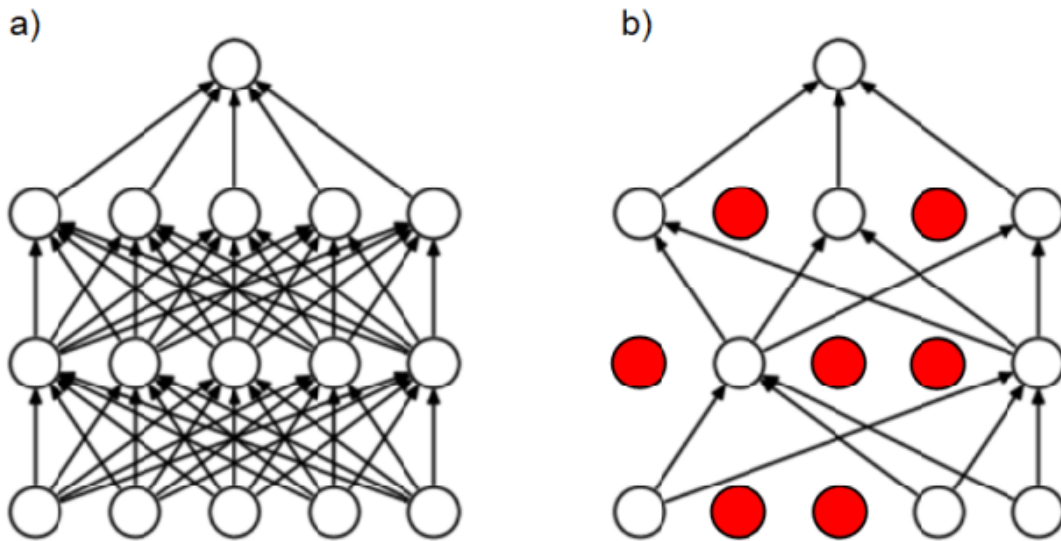
# Propriedades Matemáticas

→ A operação de **MaxPooling** retira o maior elemento de determinada região da *matrix*. Posteriormente, é feito um deslizamento considerando um parâmetro de *stride* (similar a operação de convolução) para aplicação de uma nova operação.



# Propriedades Matemáticas

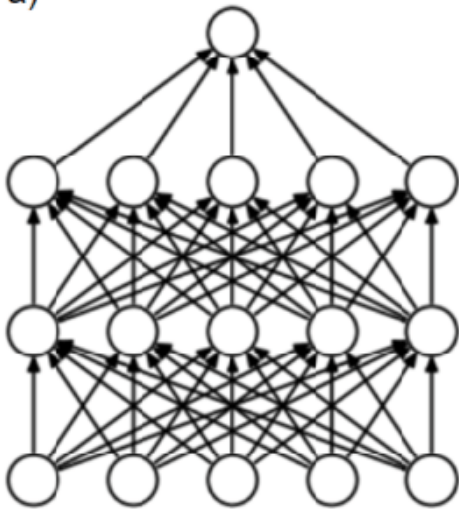
→ A camada de **Dropout** é utilizada para evitar que determinadas partes da rede neural tenham muita responsabilidade e consequentemente, possam ficar muito sensíveis a pequenas alterações.



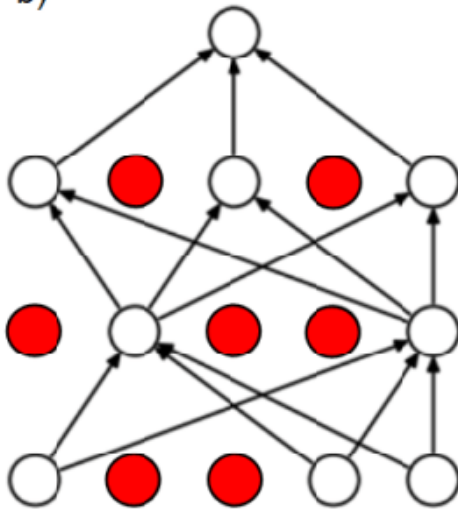
# Propriedades Matemáticas

→ Essa camada recebe um hyper-parâmetro que define uma probabilidade de “desligar” determinada área da rede neural durante o processo de treinamento.

a)



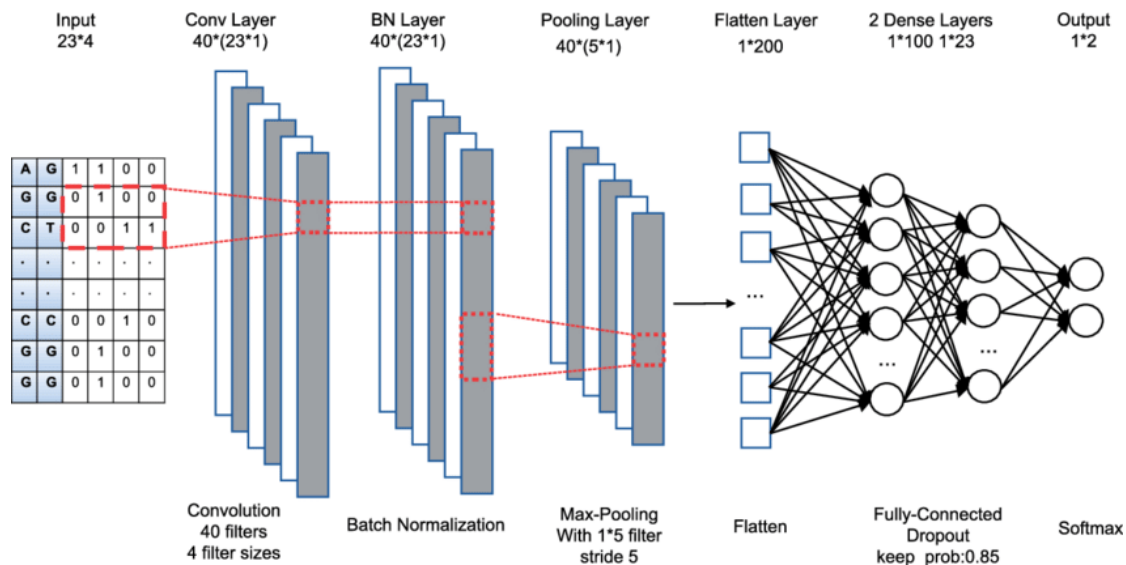
b)





# Propriedades Matemáticas

→ **Flatten**: Essa camada normalmente é utilizada na divisão das 2 partes da CNN (extração de características / rede neural tradicional). Ela basicamente opera uma transformação na *matrix* da imagem, alterando seu formato para um array.



# Obrigado!

*Machine Learning*

Prof. Dr. Diego Bruno

