

# APRENDIZADO DE MÁQUINA

REDES CONVOLUCIONAIS

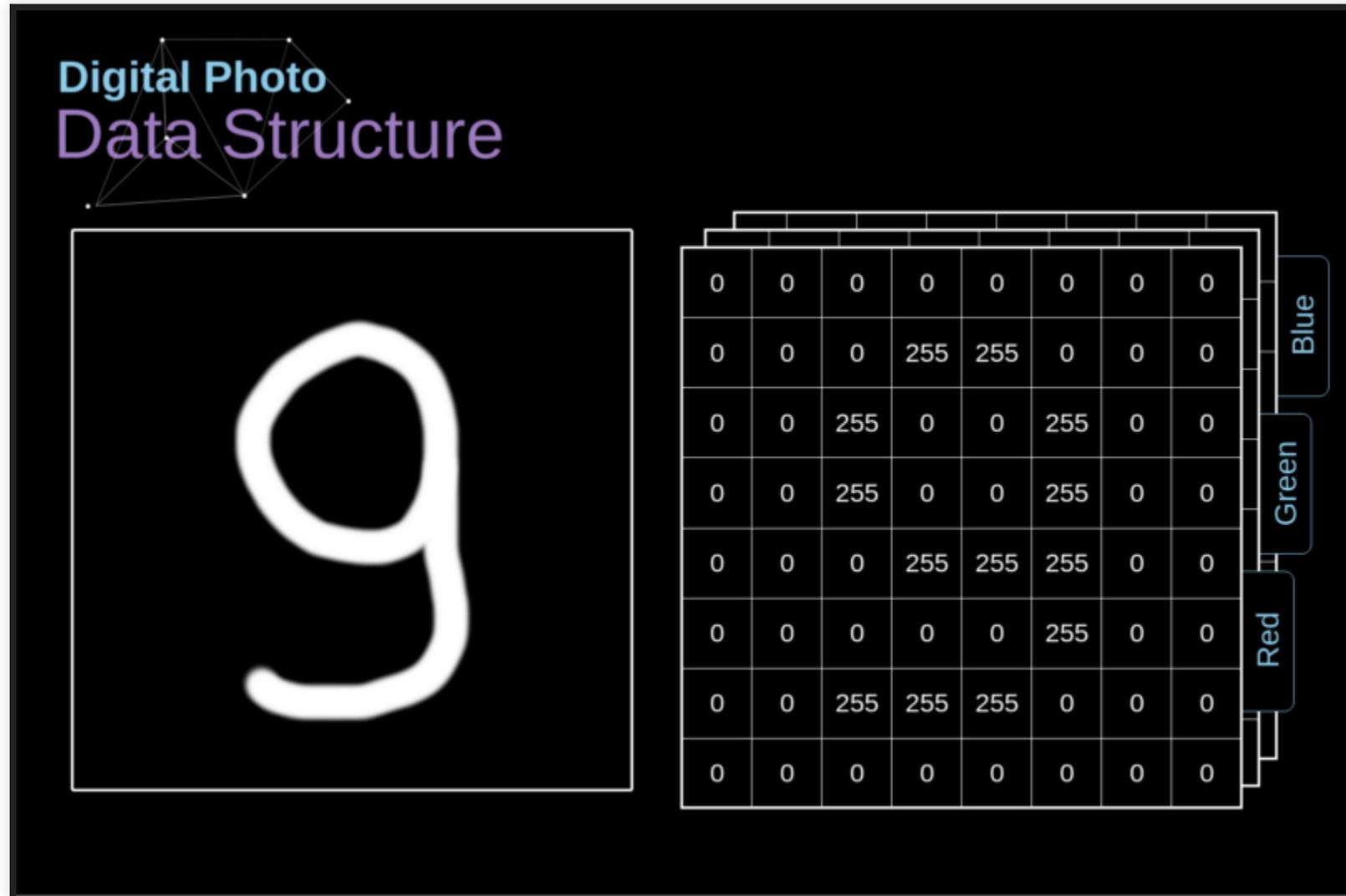
PROF. RONALDO CRISTIANO PRATI

[ronaldo.prati@ufabc.edu.br](mailto:ronaldo.prati@ufabc.edu.br)

Bloco A, sala 513-2



# REPRESENTAÇÃO DE IMAGEM



# REDE NEURAL DENSAMENTE CONECTADA

- Em muitos problemas, como imagens ou séries temporais, o tamanho da entrada é enorme:
  - Um celular com uma resolução de 12MP tem 36M de elementos
  - Uma rede neural com uma única camada de 100 neurônios teria 3.6 bilhões de parâmetros

# REDE NEURAL DENSAMENTE CONECTADA

- Quantidade grande de parâmetros
  - risco de overfitting
- Inadequadas a imagens de alta resolução
  - requer muitas imagens
- Tempo para computar as ativações/pré-ativações.

# REDE NEURAL DENSAMENTE CONECTADA

- Ignoram a estrutura espacial existente em imagens
- Se as imagens são linearizadas os pixels próximos e os localizados em regiões distantes tratados indistintamente.
- A própria rede teria que detectar as dependências existentes na estrutura espacial da distribuição próximas às imagens de entrada.

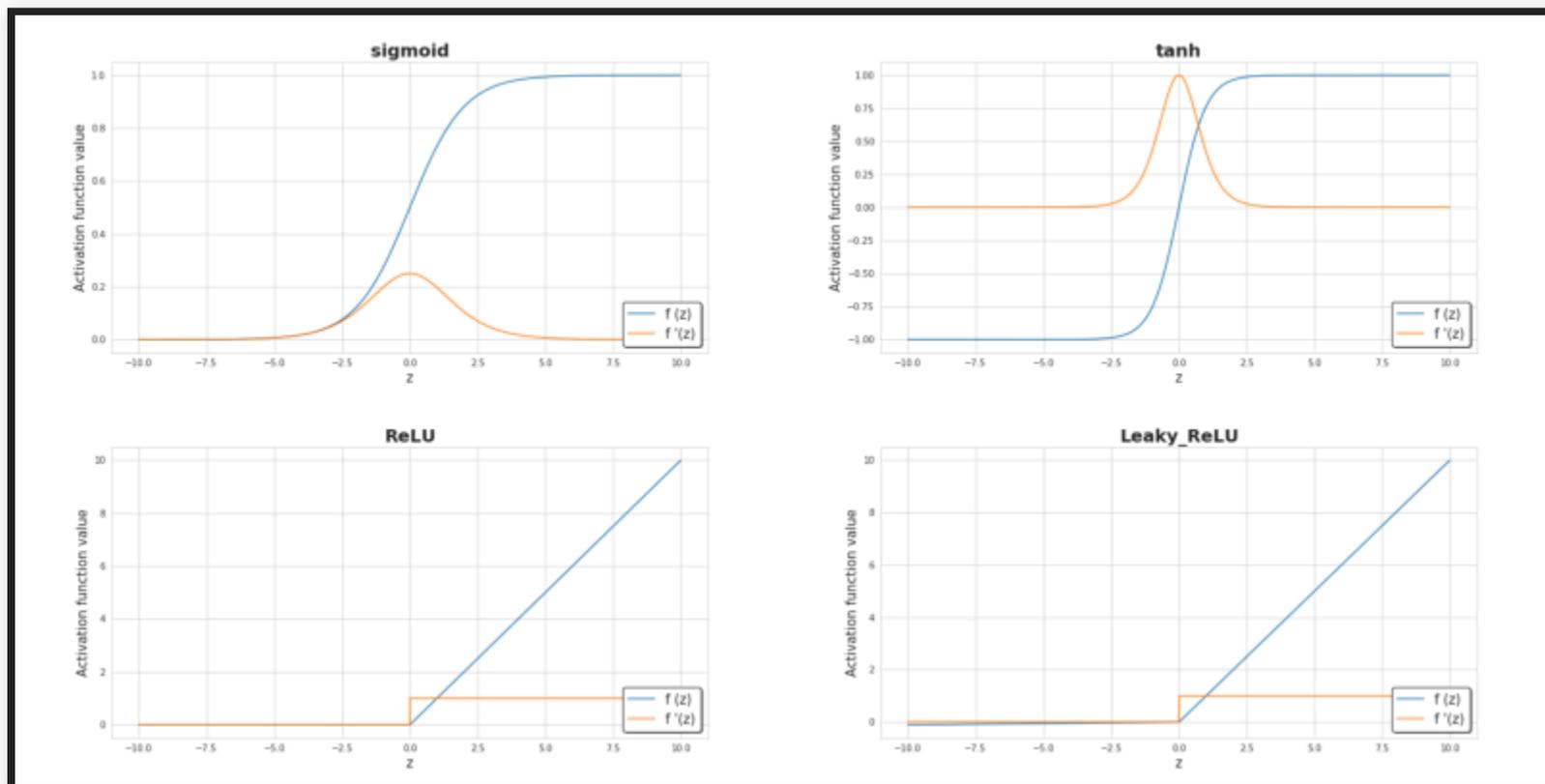
# INSTABILIDADE DO GRADIENTE

- Uma rede desse tamanho poder ter problemas para convergir
  - "Desaparecimento do gradiente": gradiente fica muito pequeno (muito próximo de zero) - muito tempo e muitos dados para treinar
  - "Explosão do gradiente": gradiente muito grande que não converge

# FUNÇÃO DE ATIVAÇÃO

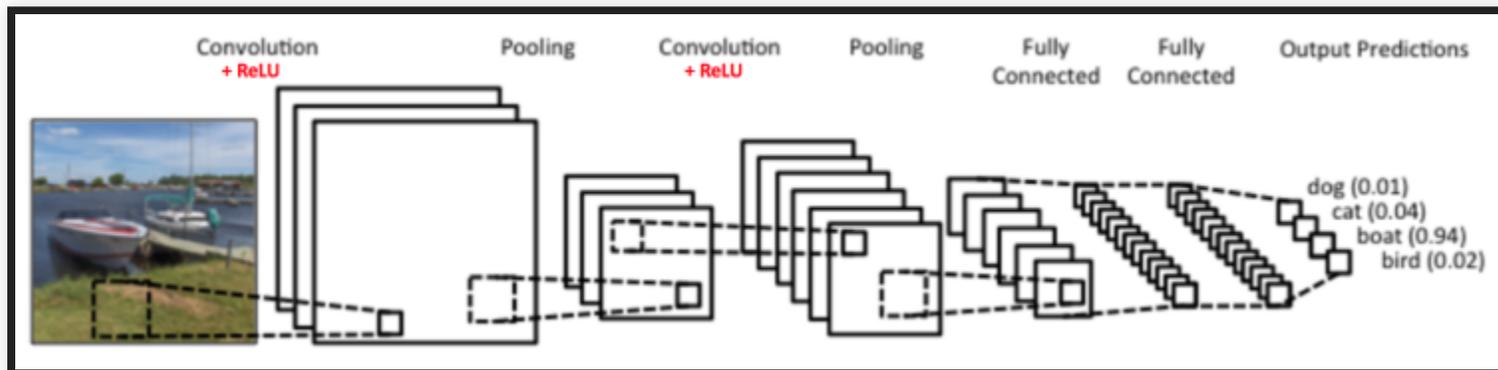
- Usar outras funções pode ajudar a evitar evitar o desaparecimento do gradiente
- A função de ativação RELU (rectified linear unit) evita o desaparecimento do gradiente
- Entretanto, pode "matar" a entrada (maioria das saídas é igual a zero)

# FUNÇÃO GRADIENTE



# REDES CONVOLUCIONAIS (CNN)

- Uma CNN é adaptada para explorar a correlação espacial existente em imagens
- Inspirada no funcionamento do cérebro (córtex visual).



# CONCEITOS E OPERAÇÕES

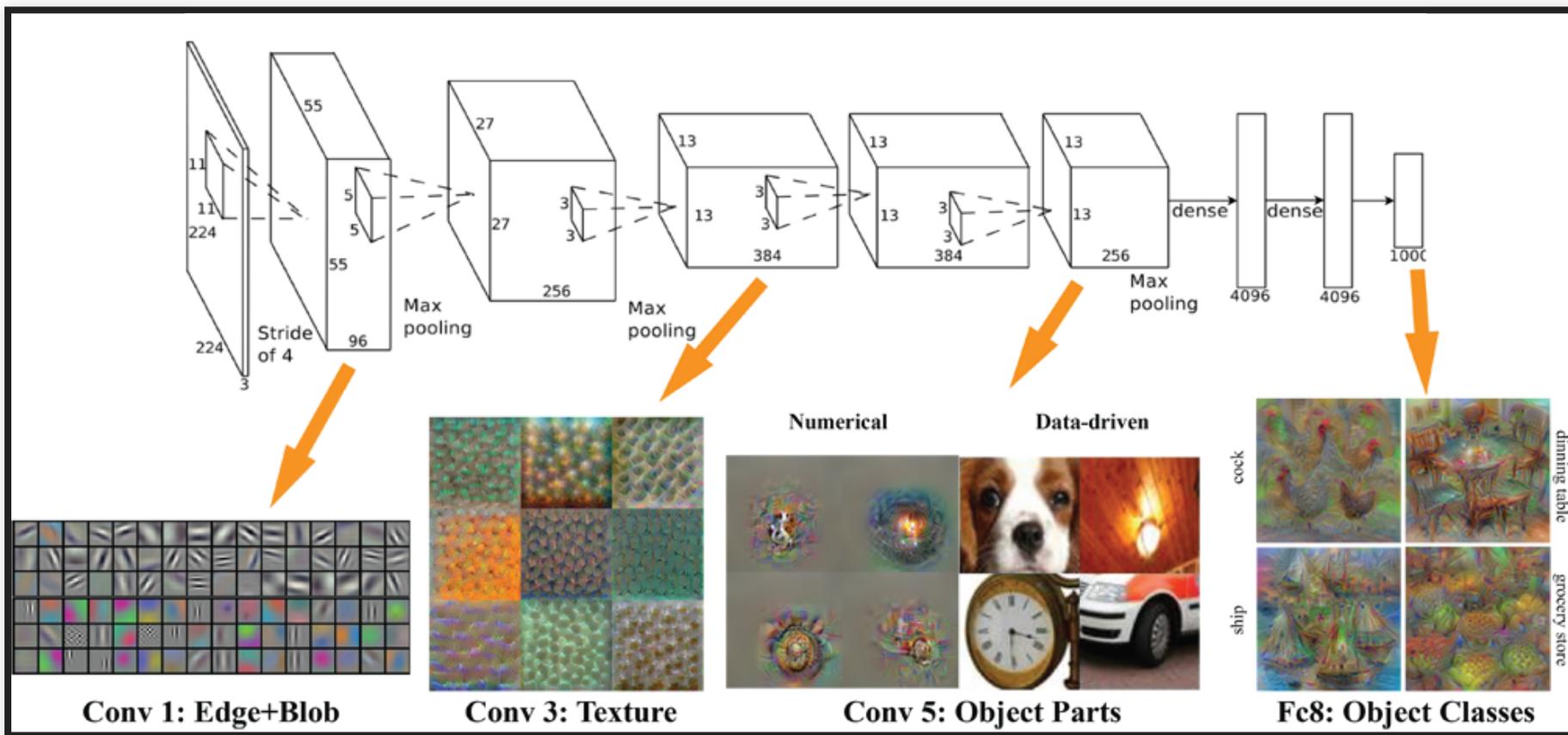
- Conceitos:
  - campos receptivos locais (local receptive fields)
  - mapas de características (feature maps, activation maps)
  - compartilhamento de pesos (shared weights).
- Operações
  - convolução (convolution)
  - subamostragem (subsampling, pooling)
  - zero-padding

# CAMPOS RECEPTIVOS LOCAIS

- Neurônios em uma CONV utilizam conectividade local (em vez de conectividade global)
  - cada neurônio está conectado a uma região contígua de unidades da camada anterior.
- Essa região é o campo receptivo local dessa unidade.

# CAMPOS RECEPTIVOS LOCAIS

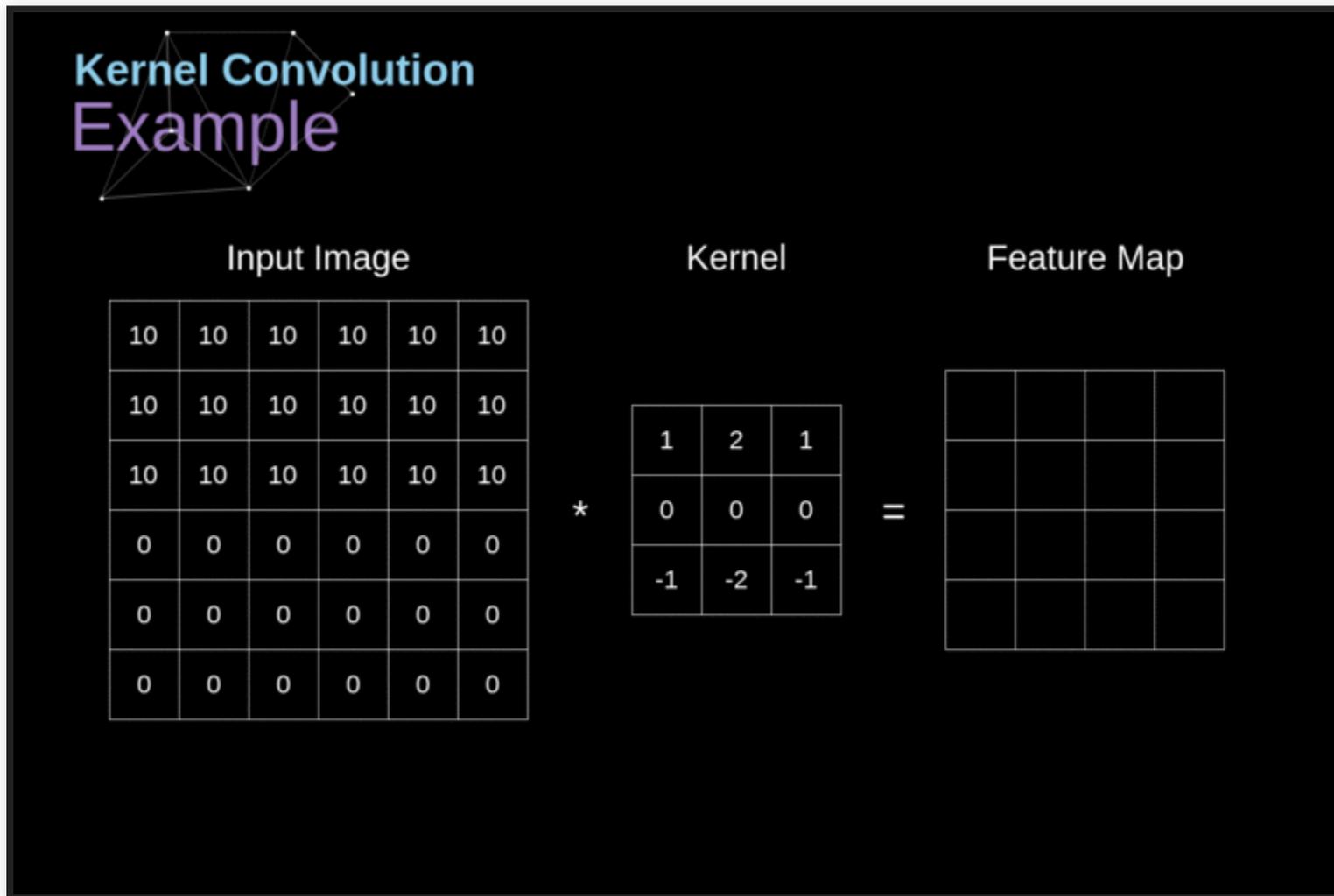
- Por meio de seu campo receptivo, cada neurônio pode detectar **características visuais elementares**
  - (e.g., arestas orientadas, extremidades, cantos)
- ... que podem então ser combinadas por camadas subsequentes para detectar **características visuais mais complexas**
  - (e.g., olhos, bicos, rodas, etc.)



# OPERAÇÃO DE CONVOLUÇÃO

- O tipo de camada principal em uma CNN é a camada de convolução (convolution layer, CONV).
- Cada neurônio em uma CONV aplica um operador de convolução (filter, kernel) sobre seu campo receptivo.
- Um operador de convolução é uma matriz!
  - Objetivo: extrair características(features) da imagem de forma automática durante o treinamento.

# OPERAÇÃO DE CONVOLUÇÃO



# OPERAÇÃO DE CONVOLUÇÃO

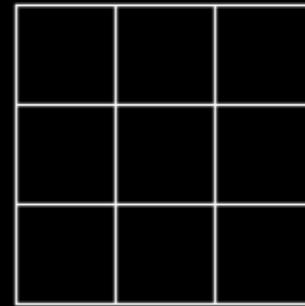
- *Analogia:* a convolução corresponde a mover uma lanterna da direita para a esquerda, e de cima para baixo, até chegar ao canto inferior direito da imagem de entrada.
- A cada região iluminada, o filtro da convolução é aplicado na tentativa de detectar alguma característica visual.

# OPERAÇÃO DE CONVOLUÇÃO

- No contexto de uma CNN, a aplicação de uma convolução corresponde a computar o produto escalar (dot product) entre a entrada e o filtro.
- A definição geral é mais complexa...
- Resultado: outra matriz!
  - mapa de ativação (activation map) ou mapa de características (feature map)

# OPERAÇÃO DE CONVOLUÇÃO

# Edge Detection Using Kernel Convolution



# OPERAÇÃO DE CONVOLUÇÃO

# Edge Detection Using Kernel Convolution



1	2	1
0	0	0
-1	-2	-1

# OPERAÇÃO DE CONVOLUÇÃO

# Edge Detection Using Kernel Convolution



0	1	2
-1	0	1
-2	-1	0

# OPERAÇÃO DE CONVOLUÇÃO

# Edge Detection Using Kernel Convolution



1	0	-1
0	0	0
-1	0	1

# OPERAÇÃO DE CONVOLUÇÃO

Edge Detection

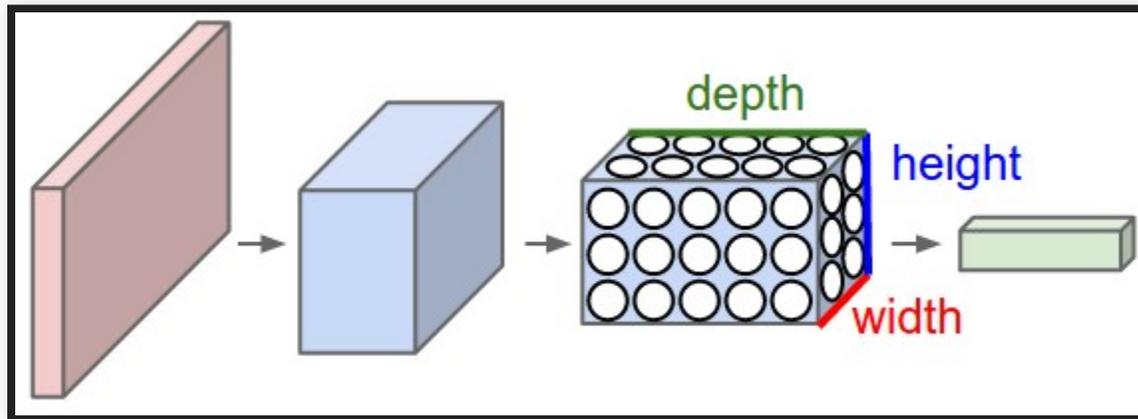
Using Kernel Convolution



0	-1	0
-1	5	-1
0	-1	0

# CONVS

- Uma CONV é um arranjo tridimensional (tensor) de neurônios: altura, largura, profundidade.

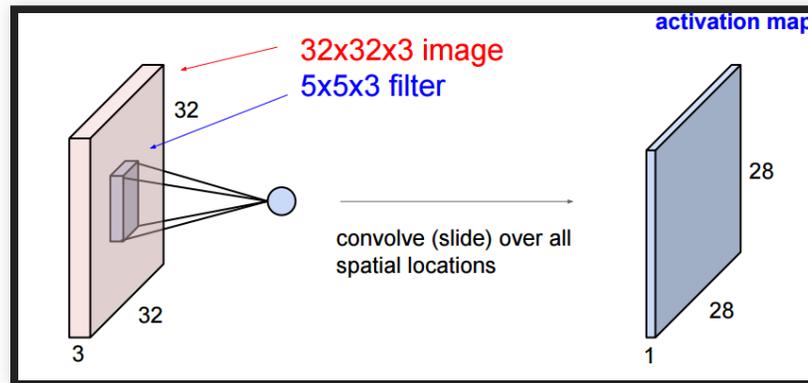


# CONVS

- Os neurônios em uma CONV aplica um filtro (operação de convolução) ao seu campo receptivo
- Neurônios de uma mesma camada de profundidade aplicam o mesmo filtro.
- O objetivo de cada filtro é ativar quando detecta um tipo particular de característica na entrada.

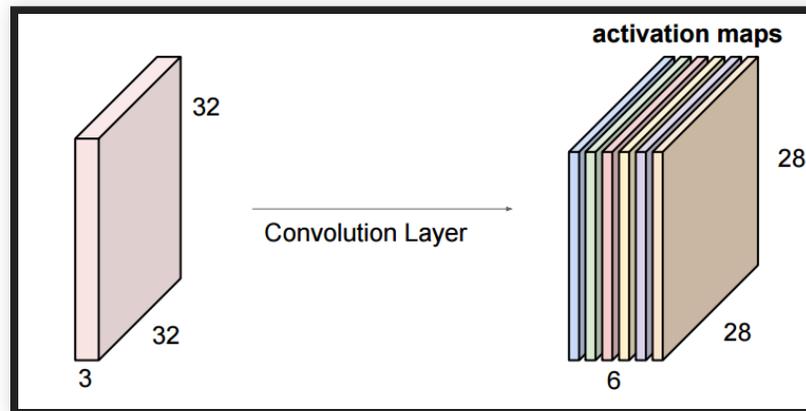
# CONVS

- Durante a propagação adiante (forward pass), cada filtro é aplicado (convolved) através do volume de entrada.



# CONVS

- O volume de saída de uma CONV é uma sequência de mapas de ativação
- A sequência de mapas de ativação de um volume forma a dimensão de profundidade (depth) desse volume.

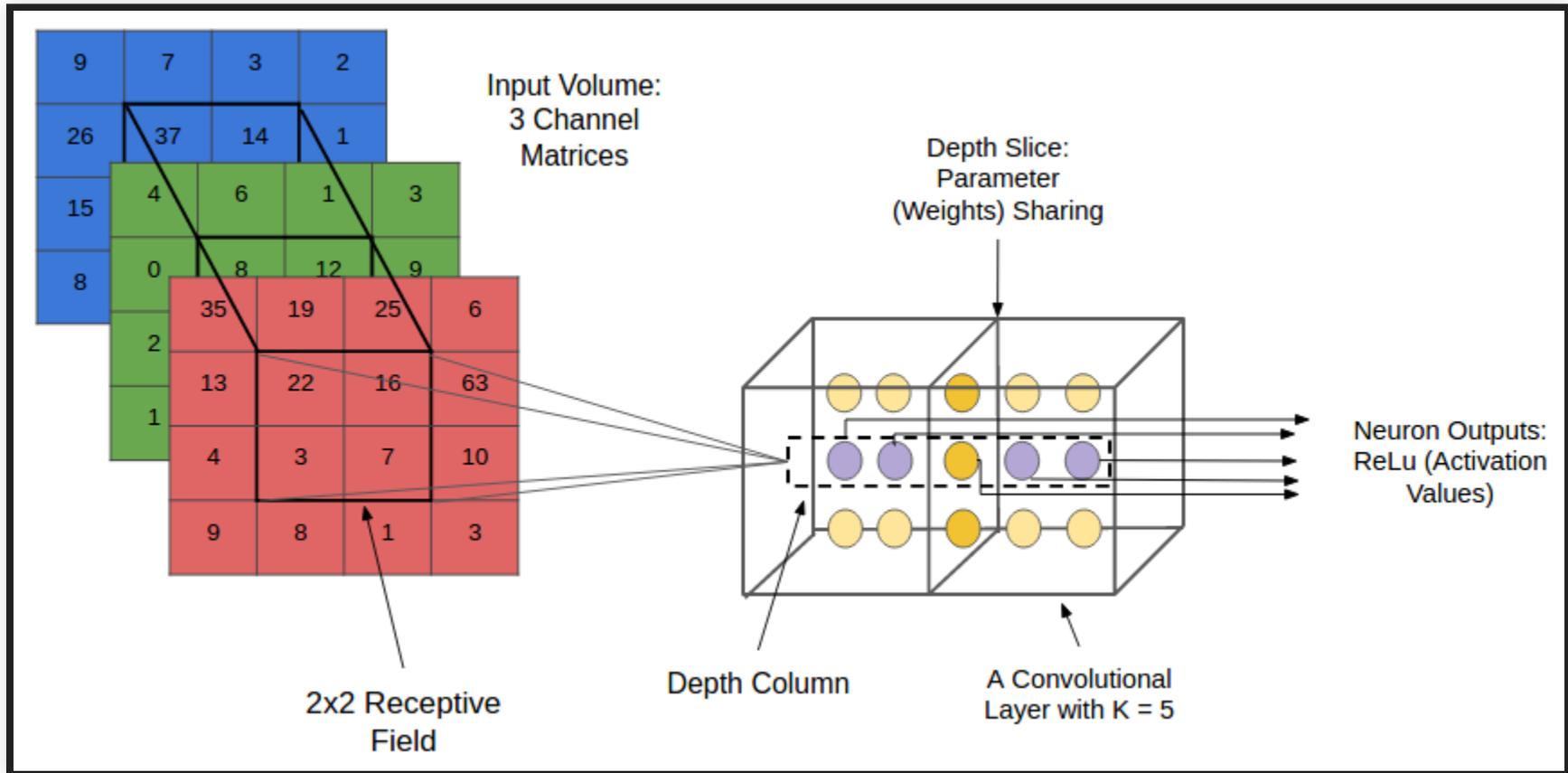


**CONVS**

# CONVS

- Cada entrada em um volume de saída é resultante da computação de um neurônio que analisa uma pequena região na entrada e
- compartilha parâmetros com neurônios no mesmo mapa de ativação.

# CONVS



# CONVS

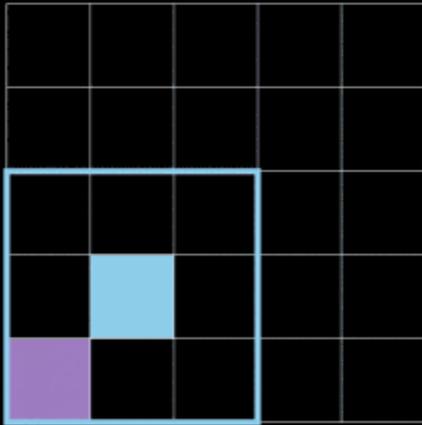
- As convoluções utilizam de maneira diferente os elementos que estão nos cantos, bordas e no meio da imagem
- Os elementos dos cantos e bordas podem ser "subutilizados"

# CONVS

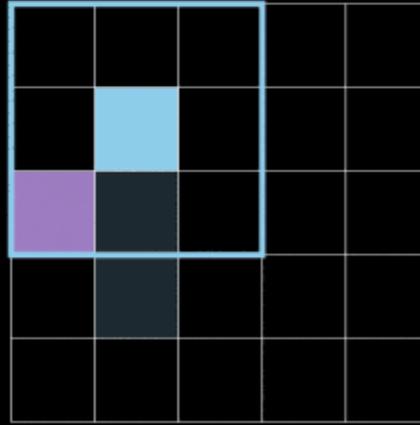
## Impact of Pixel Position



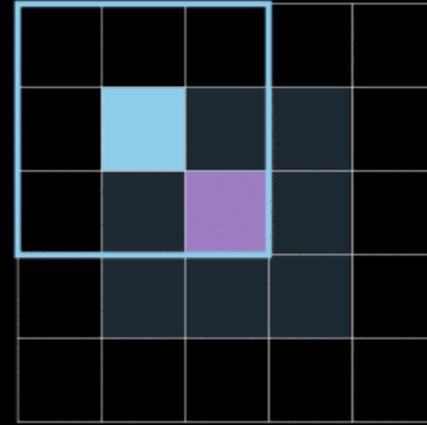
Corner



Ende



Inside

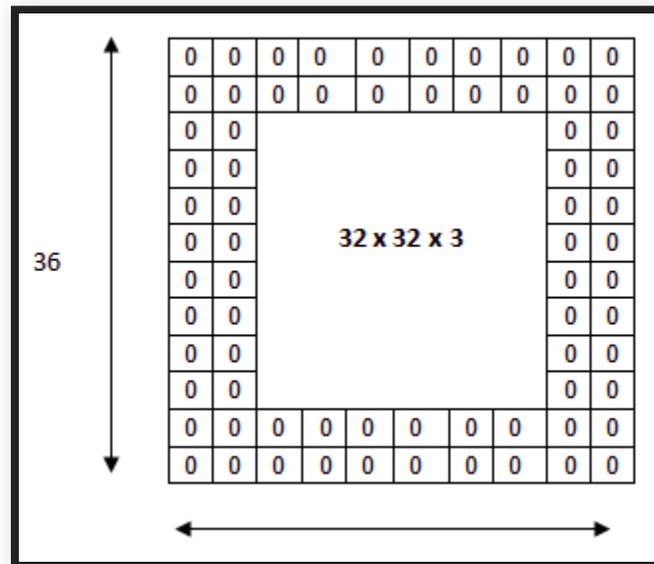


# ZERO PADDING

- Corresponde a adicionar simetricamente zeros a cada fatia (depth slice) do volume de entrada.
- Aumenta as dimensões da entrada.

# ZERO PADDING

Uma imagem com 32x32 pixels é preenchida com zeros para aplicar uma máscara de 3x3



# STRIDE

- O passo (stride,  $S$ ) é um hiperparâmetro que controla de que forma o filtro é “deslizado” pelo volume de entrada durante a convolução

# STRIDE

## Strided Convolution

Input Image

1	9	8	4	4	5	7
4	8	6	7	9	1	7
4	0	5	9	3	8	4
7	3	5	9	0	5	4
7	4	1	1	8	1	2
7	6	6	9	8	7	6
3	6	3	5	4	2	7

Kernel

0	-1	0
-1	5	-1
0	-1	0

\*

=

Feature Map


# CAMADA DE POOLING

- Por vezes encontrada entre duas camadas CONVs sucessivas para reduzir a complexidade do modelo (mitigar overfitting).
- **Objetivo:** reduzir dimensões espaciais (largura x altura) do volume de entrada para a próxima CONV.
- Aplicada a cada fatia de profundidade do volume de entrada.
- Profundidade do volume de saída igual à do volume de entrada.

# CAMADA DE POOLING

## Max Pooling Example



Input

3	0	1	5	1	3
5	7	3	4	4	6
7	7	1	8	3	5
6	1	7	0	0	5
0	4	5	5	7	2
3	2	0	2	0	2

Output


# CAMADA DE POOLING

