

ANA RAQUEL

POSTECH

DATA ANALYTICS

DEEP & REINFORCEMENT LEARNING

# AULA 02

## SUMÁRIO

O QUE VEM POR AÍ? .....	3
HANDS ON .....	4
SAIBA MAIS .....	5
O QUE VOCÊ VIU NESTA AULA? .....	17
REFERÊNCIAS.....	18

EMANDA

## O QUE VEM POR AÍ?

Você sabia que as redes neurais conseguem identificar e distinguir imagens, assim como nós (seres humanos)? Nessa aula aprenderemos como funciona a arquitetura das **Redes Neurais Convolucionais (CNN)**! As CNNs emergiram do estudo do córtex visual cerebral e têm sido utilizadas no reconhecimento de imagens desde a década de 80. Então, vamos aprender como funciona a arquitetura dessa rede e entender o motivo dela ser tão poderosa?



## HANDS ON

Vamos aprender a criar um classificador de imagens? Nas próximas aulas, veremos o passo a passo de como criar uma arquitetura de redes convolucionais aplicando reconhecimento de imagens.

O código desta aula você encontra aqui: [código da aula 02](#).

Além disso, também temos a [base de dados da aula 02](#) e [imagens para teste](#).

EMANSP

## SAIBA MAIS

### ENTENDENDO A INSPIRAÇÃO DAS REDES NEURAIS CONVOLUCIONAIS

Assim como aprendemos nas redes neurais multicamadas, as CNNs também são inspiradas em fatores biológicos. Em 1958 e 1959, David H. Hubel e Torton Wiesel realizaram uma série de experimentos com gatos, na qual proporcionaram o aprendizado sobre a estrutura do córtex visual.

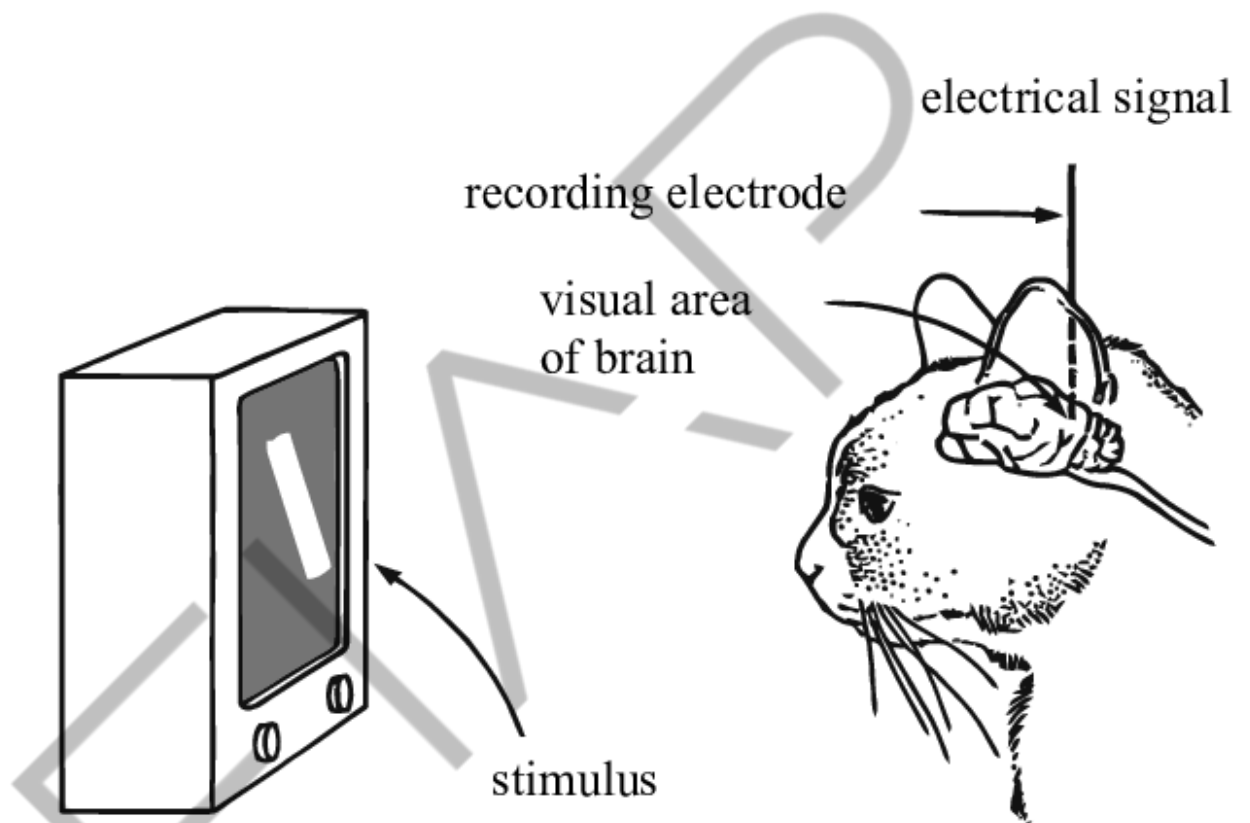


Figura 1 - Experimento realizado no córtex visual em um gato.

Fonte: Anh Nguyen (2019)

Hubel e Wiesel descobriram que os neurônios do córtex visual de um gato **são ativados juntos quando expostos a algumas linhas ou curvas**. Inspirado no córtex visual, saber distinguir diferentes imagens é possível quando **identificamos e comparamos diferentes características em cada ponto específico na imagem**. Eles também notaram que alguns neurônios têm campos receptivos maiores e reagem a padrões mais complexos que são combinações dos padrões de nível inferior. Essas observações levaram à ideia de que os neurônios de nível superior são baseados nas

saídas dos neurônios vizinhos de nível inferior. Esta poderosa arquitetura é capaz de detectar todos os tipos de padrões complexos em qualquer área do campo visual.

### Como o computador identifica imagens

Você deve estar se perguntando: “Como o computador consegue identificar uma imagem”, certo? Sabemos que trabalhar com dados não estruturados pode ser uma tarefa não tão simples. As entradas das redes CNNs são, basicamente, imagens, o que torna o seu input de dados diferenciado. Sabemos que o computador entende números, correto? Pois bem, as redes neurais basicamente recebem uma **matriz de pixels da imagem como entrada da rede**.

Existem os tipos de imagens em escala de cinza (grayscale) e as imagens coloridas (RGB). Essas imagens são compostas por pixels que variam entre 0 a 255, mostrando a intensidade de cores. A imagem a seguir, por exemplo, está representada por uma matriz 2D, onde cada posição da imagem é composta por elementos que variam de 0 (preto) a 255 (branco). Quanto mais próximo de 255, mais clara é a cor em escala de cinza.

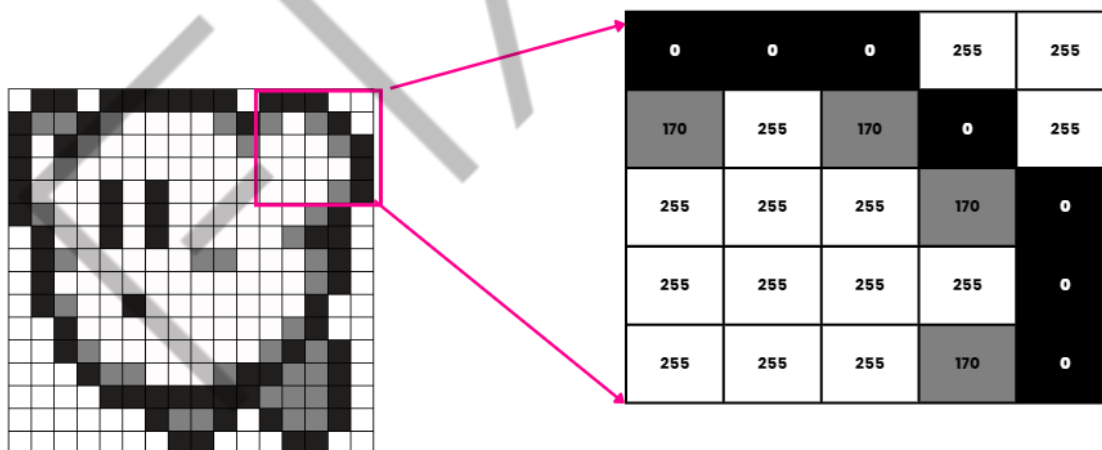


Figura 2 - Pixels de uma imagem grayscale.

Fonte: elaborado pela autora (2023)

Já as imagens em escala colorida (RGB), são representadas por uma matriz 3D, combinando as cores vermelho, verde e azul. O propósito principal do sistema

RGB é a reprodução de cores em dispositivos eletrônicos como monitores de TV, computadores, mídias digitais e celulares em geral.

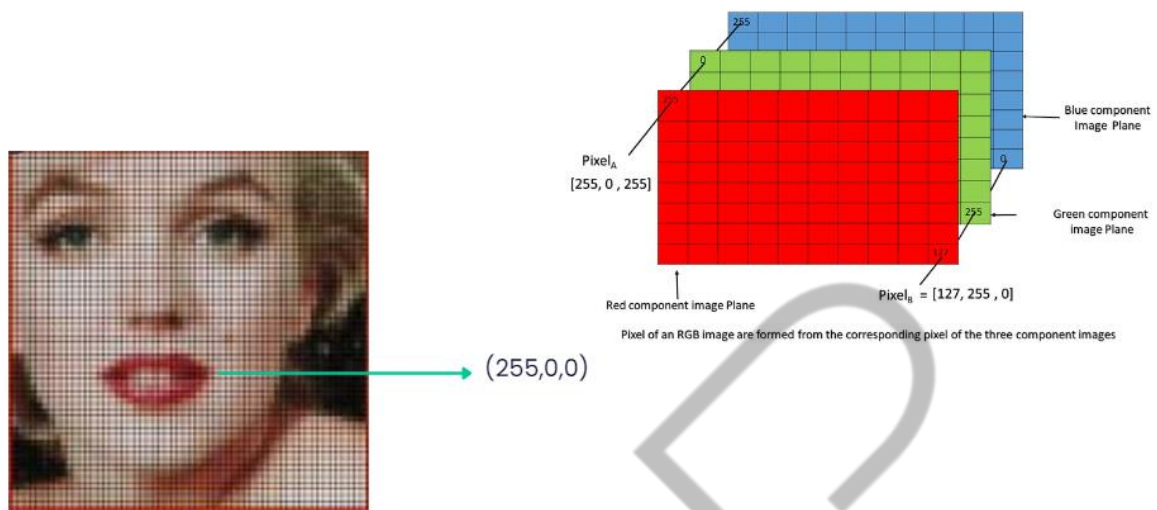


Figura 3 - Pixels de uma imagem rgb.

Fonte: elaborado pela autora (2023)

### Arquitetura das redes neurais convolucionais

Quando pensamos em redes CNN, a **camada convolucional** é a mais importante e que faz total diferença entre as redes neurais comuns. Os neurônios na primeira camada convolucional não estão conectados a cada pixel na imagem de entrada, assim como a rede que aprendemos na última aula, mas apenas a pixels em campos respectivos. Desta forma, cada neurônio da segunda camada convolucional está conectado apenas a neurônios localizados dentro de um pequeno retângulo na primeira camada. Elas são responsáveis pela etapa de aprendizagem das partes da imagem como um todo.

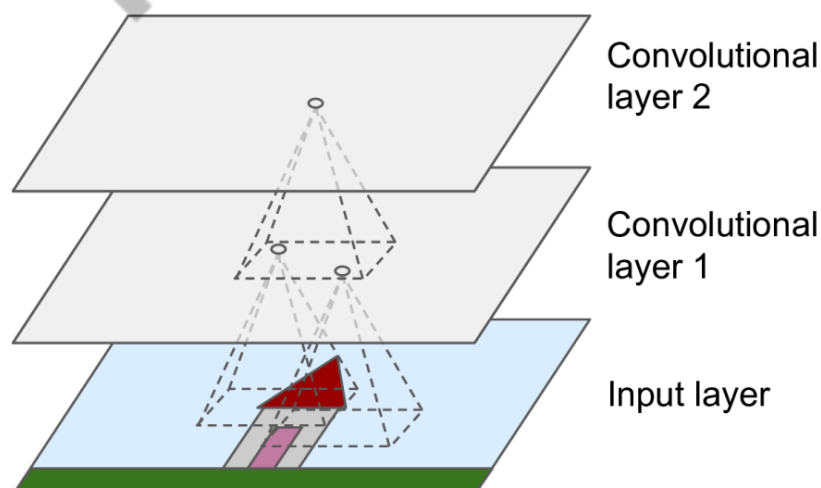


Figura 4 - Camadas NN com campos receptivos locais retangulares.  
Fonte: Aurélien Géron (2019)

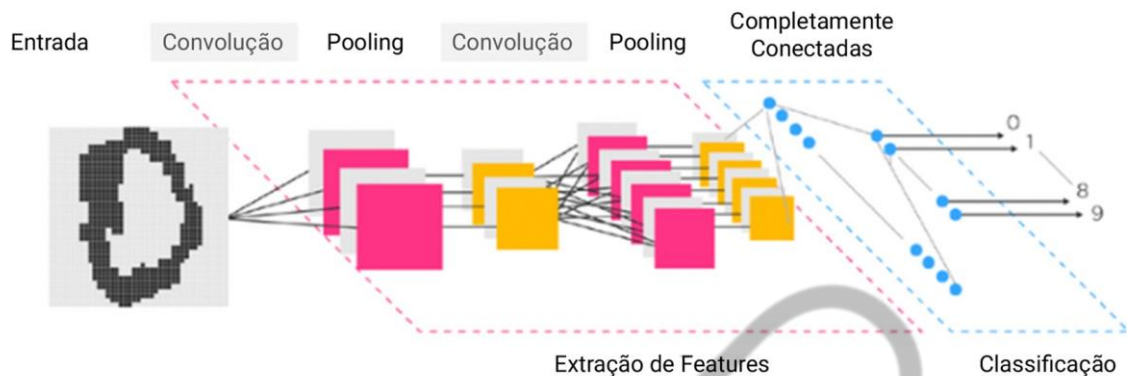
Para entendermos de forma simples o funcionamento das redes convolucionais, utilizaremos um exemplo clássico nos estudos das redes neurais: imagine que a rede neural convolucional será treinada para classificar dígitos escritos a mão em números de 0 até 9. Cada pessoa possui uma caligrafia única e nem sempre as escritas são iguais. Um grande desafio para a rede, certo?



Figura 5 - Amostra de dígitos escritos à mão.  
Fonte: Analytics Vidhya (2021)

Vamos pegar um exemplo do dígito zero (0). Imagine que a figura 5 será a entrada da rede e a arquitetura da rede tem a estrutura da figura 6.





Fonte: Neto, 2017 - Aprendizado profundo: conceitos, técnicas e estudo de caso de análise de imagens com Java

Figura 6 - Arquitetura de redes neurais convolucionais.  
Fonte: Francisco Gerardo Medeiros Neto, et. al. (2017)

Observe que temos a primeira camada chamada de **convolução**, que é responsável pelo papel de "**detector de atributos**". Essa etapa vai mapear a região de uma imagem para, posteriormente, termos um "**mapa de atributos**". Funciona basicamente como um filtro, na qual chamamos de **kernel** na rede. O filtro nada mais é do que a multiplicação de matrizes, e esse processo é realizado diversas vezes por toda a imagem recebida.

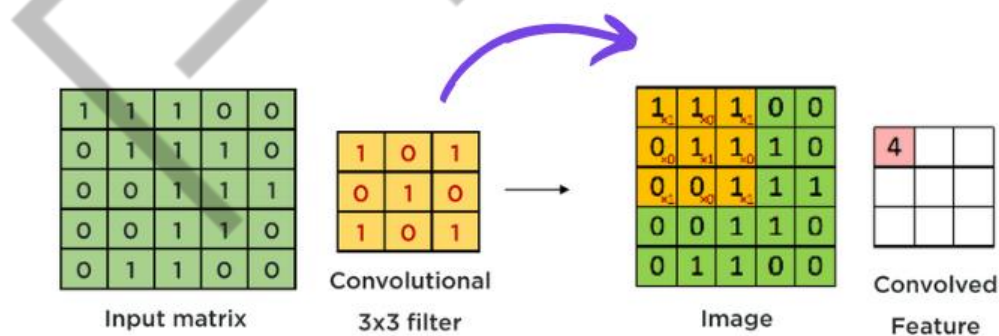


Figura 7 - Camadas convolucionais.  
Fonte: elaborado pela autora (2023)

Ao realizar a convolução, o mapa de atributos começa a aprender as partes importantes para identificar a composição da imagem (assim como David H. Hubel e Torton Wiesel concluíram que os neurônios do córtex visual de um gato são ativados

juntos quando são expostos a algumas linhas ou curvas). Cada pixel será identificado dentro da etapa de convolução, passando pelo filtro kernel movido por um multiplicador, realizando assim a transformação da imagem em um formato compactado.

Essa transformação acontece para a rede mapear quais são os pontos “chaves” que caracterizam essa imagem e armazena em uma representação dimensionalizada para passar para a próxima etapa de convolução da rede.

O multiplicador dessa etapa de convolução se chama “**stride**” (**deslocamento**) (normalmente com o valor = 1), cuja principal função é realizar a **transformação** da imagem. Toda essa transformação irá resultar em uma imagem com uma dimensão menor.

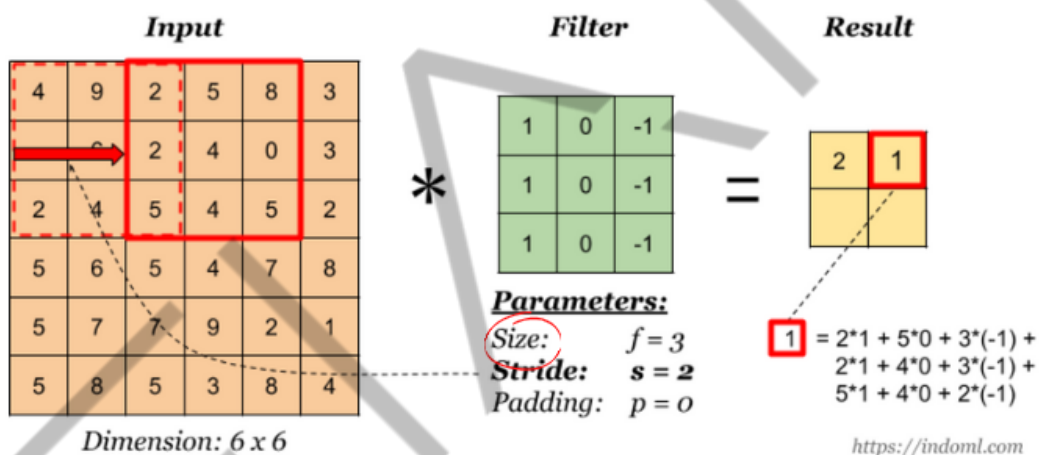


Figura 8 - Deslocamento com stride nas imagens.

Fonte: <https://indoml.com>.

É comum adicionar zeros ao redor das imagens de entrada, para que uma camada tenha a mesma altura e largura da camada anterior. Esse comportamento é chamado de **padding** na rede. O objetivo dele é manter a dimensionalidade da imagem. Na configuração das redes neurais convolucionais, podemos parametrizar o padding como “**same**” ou “**valid**”.

- **Valid** significa que a rede não criará a borda de zeros na imagem e pode ignorar algumas linhas e colunas na parte inferior e direita da imagem de entrada.
- **Same** significa que será criada a borda de zeros na imagem.

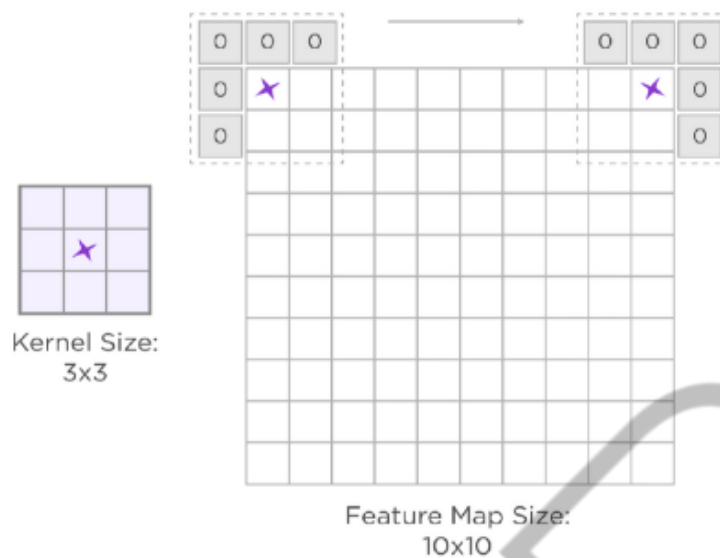


Figura 9 - Padding.  
Fonte: elaborado pela autora (2023)

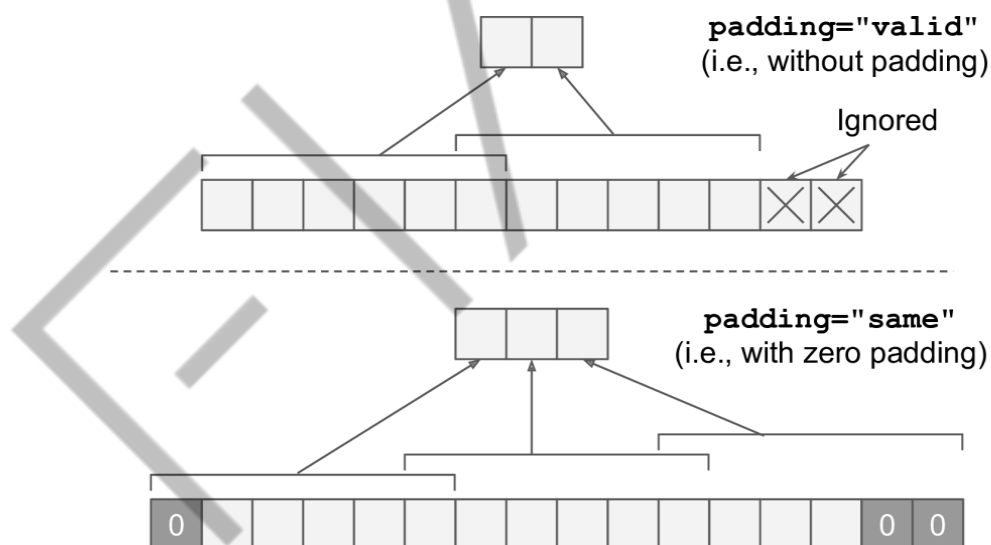


Figura 10 – Padding 2  
Fonte:Fonte: Aurélien Géron (2019)

Imagine que a imagem do dígito zero entre na primeira camada convolucional, com um tamanho de 32x32x3, ou seja, a altura e largura são iguais a 32, e o 3 significa a escala de cores da imagem, que no caso está em grayscale, então temos pixels que variam de 0 a 255.



Figura 11 - Exemplo da camada convolucional.  
Fonte: elaborado pela autora (2023).

Logo em seguida, essa imagem passa pelo kernel (filtro) e convolve a imagem, tornando-a em uma dimensão menor de 5X5X3.

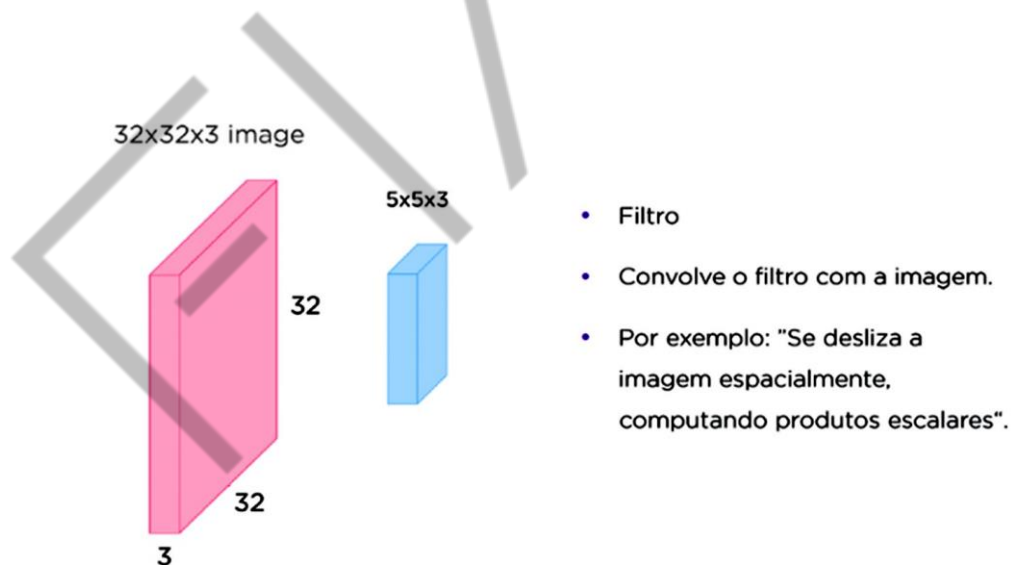


Figura 12 - Exemplo da camada convolucional.  
Fonte: elaborado pela autora (2023).

Talvez você esteja se perguntando: "e por que esses filtros dão certo?". Bem, os pesos dos neurônios podem ser representados por uma imagem menor. Durante o

treinamento, a CNN encontra os filtros mais úteis para sua tarefa de aprender a combiná-los em padrões mais complexos.

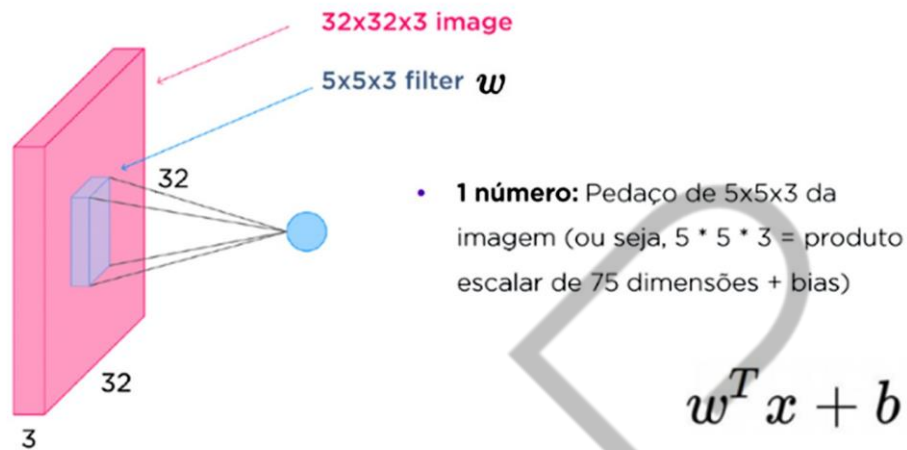


Figura 13 - Exemplo da camada convolucional com kernel.  
Fonte: elaborado pela autora (2023).

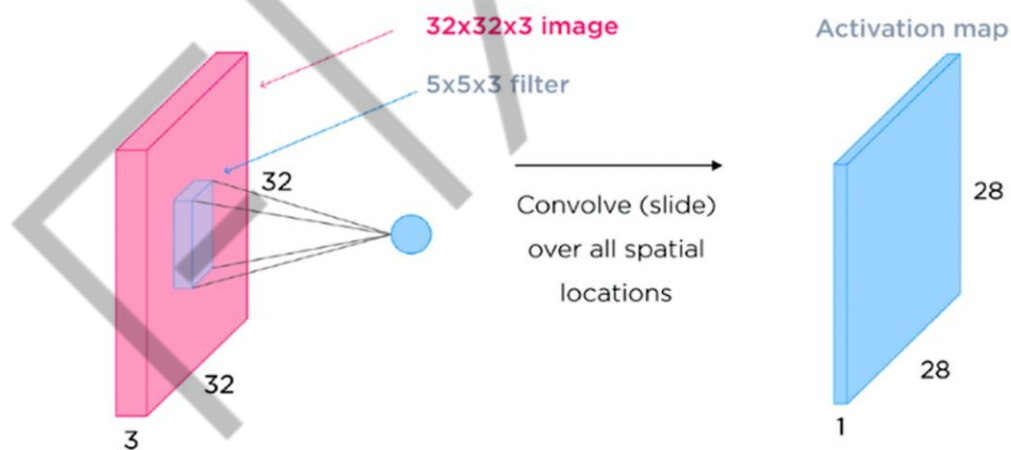


Figura 14 - Mapa de ativação criado na etapa de filtro do kernel.  
Fonte: elaborado pela autora (2023).

A cada medida que realizamos essas extrações de características das imagens, adicionamos novos mapas de filtros. Exemplo: se são 6 filtros de 5x5, teríamos 6 mapas de ativação.

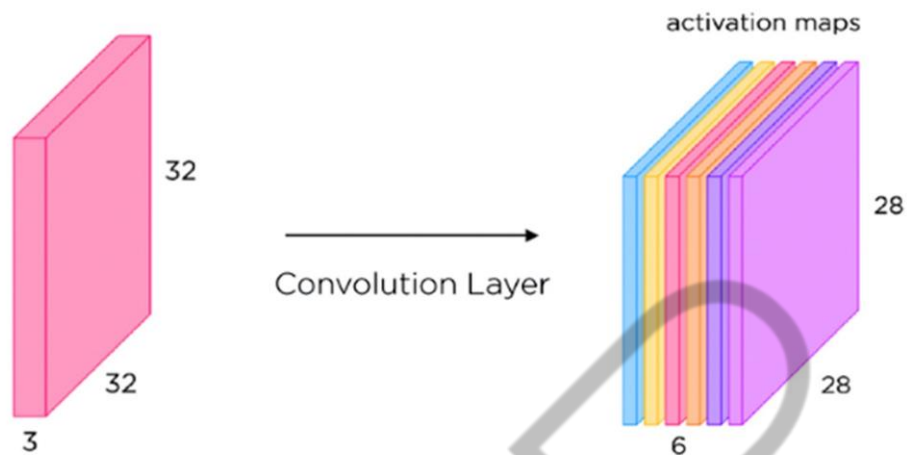


Figura 15 - Mapas de ativação.  
Fonte: elaborado pela autora (2023)

Entre as camadas de mapa de atributos, vamos ter uma **função de ativação** para realizar a classificação das imagens. Nossa, então, rede torna-se uma sequência de camadas de convolução, intercaladas por funções de ativação. A função de ativação ReLU é muito utilizada para treinar imagens devido ao seu comportamento de não ativar todos os neurônios ao mesmo tempo (isso traz mais performance para a rede).

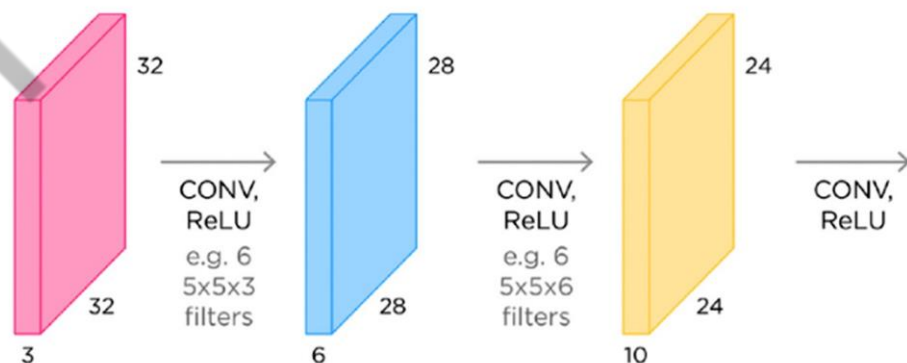


Figura 16 - Mapas de ativação entre funções de ativação.  
Fonte: elaborado pela autora (2023)

Em seguida, temos a camada de **pooling**. Seu objetivo é subamostrar (ou seja, diminuir) a imagem de entrada para reduzir a carga computacional, o uso da memória e o número de parâmetros (limitando o risco de overfitting). A operação **max pooling** retira o maior elemento de determinada região da matriz (imagem). Em seguida é realizado um novo cálculo de stride (assim como na convolução). A principal motivação dessa operação no modelo, é de diminuir sua variância a pequenas alterações e também de reduzir a quantidade de parâmetros treinados pela rede.

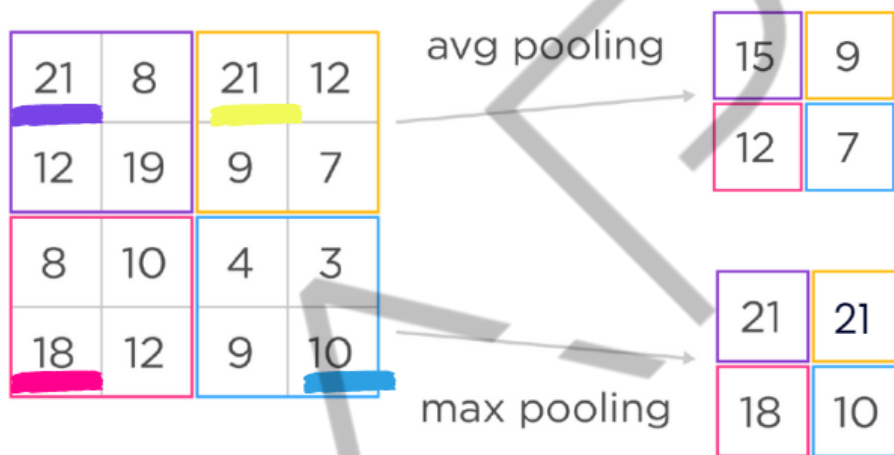


Figura 17 - Max e AVG Polling.  
Fonte: elaborado pela autora (2023)

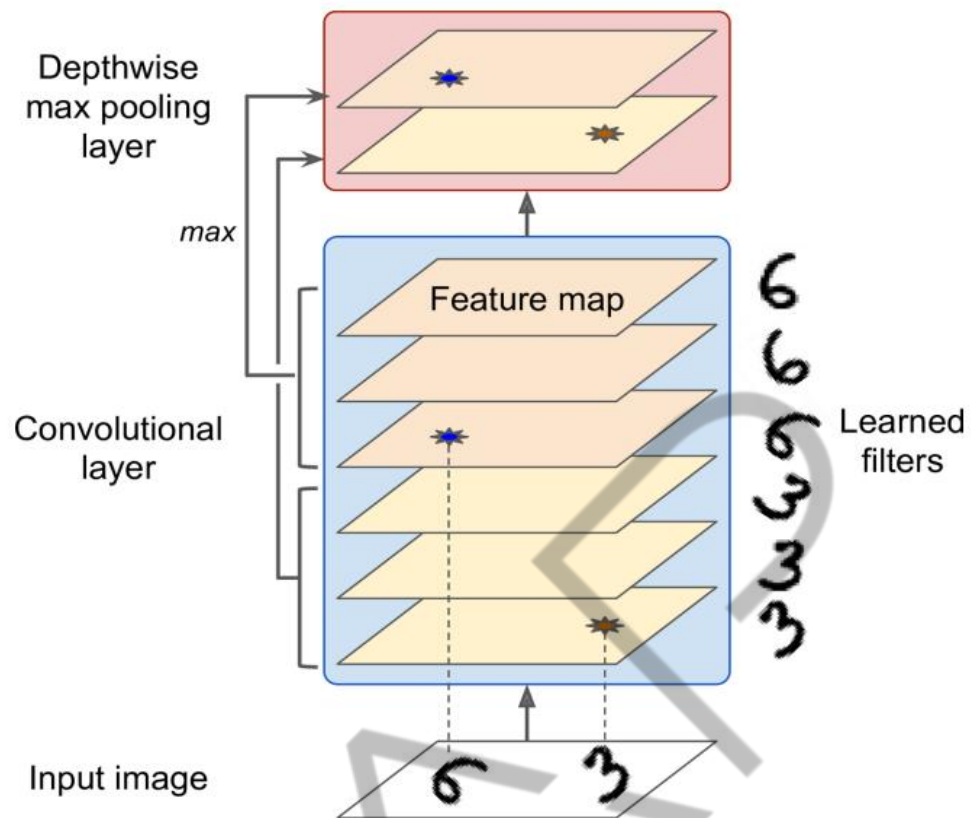


Figura 18 - O pooling máximo em profundidade pode ajudar a CNN a aprender qualquer invariância  
Fonte: Aurélien Géron (2019)

As arquiteturas CNN típicas basicamente empilham algumas camadas convolucionais com a função de ativação ReLU, depois uma camada pooling, em seguida outras camadas convolucionais com a função de ativação ReLU, depois outra camada pooling, e assim por diante. À medida em que a rede vai avançando, a imagem vai ficando cada vez menor e mais profunda (graças aos mapas de atributos nas camadas convolucionais).

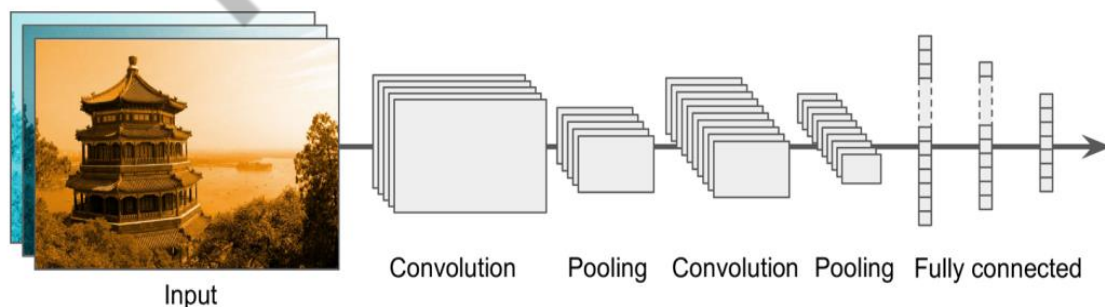


Figura 19- Arquitetura típica de CNN.  
Fonte:Fonte: Aurélien Géron (2019)



## O QUE VOCÊ VIU NESTA AULA?

Nessa aula, você aprendeu o que são as redes neurais convolucionais e como funciona a arquitetura dessa rede que lida com imagens.

O que achou do conteúdo? Conte-nos no Discord! Estamos disponíveis na comunidade para fazer networking, tirar dúvidas, enviar avisos e muito mais. Participe!

EMAND

## REFERÊNCIAS

Deep Learning Book. [s.d.]. Disponível em: <<https://www.deeplearningbook.com.br/>>.

Acesso em: 23 out. 2023.

GÉRON, Aurélien. Hands-On Machine Learning with Scikit-Learn, Keras and TensorFlow, 2nd Edition. [s.l.]: O'Reilly Media, Inc., 2019.

Why choose Keras? Keras. [s.d.]. Disponível em: <[https://keras.io/why\\_keras/](https://keras.io/why_keras/)>.

Acesso em: 23 out. 2023.

## **PALAVRAS-CHAVE**

**Palavras-chave:** Convolução, Pooling, Convolutional Layer, Kernel, Stride, ReLU.

EMSE

The background is a dark blue field filled with numerous small, light blue dots. Overlaid on this are several large, wavy, translucent lines in shades of blue and yellow. These lines flow from the left side towards the right, creating a sense of motion. Scattered throughout the composition are various geometric shapes: a thin vertical line, a circle containing the number '7', a circle containing the number '0', a small 'x' mark, and a hexagon in the lower right corner.

POS TECH