

Redes Neurais Convolucionais

Prof. Gustavo Willam Pereira

Conteúdo

- Introdução as Redes Neurais Convolucionais
- Operadores de Convolução
- Pooling e MaxPooling
- Flattening
- Rede Neural Densa

Introdução a Redes Neurais Convolucionais

- A classificação de imagens digitais é uma área do *machine learning* que chama bastante atenção.
- Como nosso cérebro consegue diferenciar, por exemplo, uma imagem uma rosa de imagem uma tulipa? Ou de um cachorro ou um gato ?
- Na verdade, nosso cérebro consegue extrair uma série de características (*features*) que possibilita realizar a classificação. Veja a figura a seguir, o que você vê?

Introdução a Redes Neurais Convolucionais

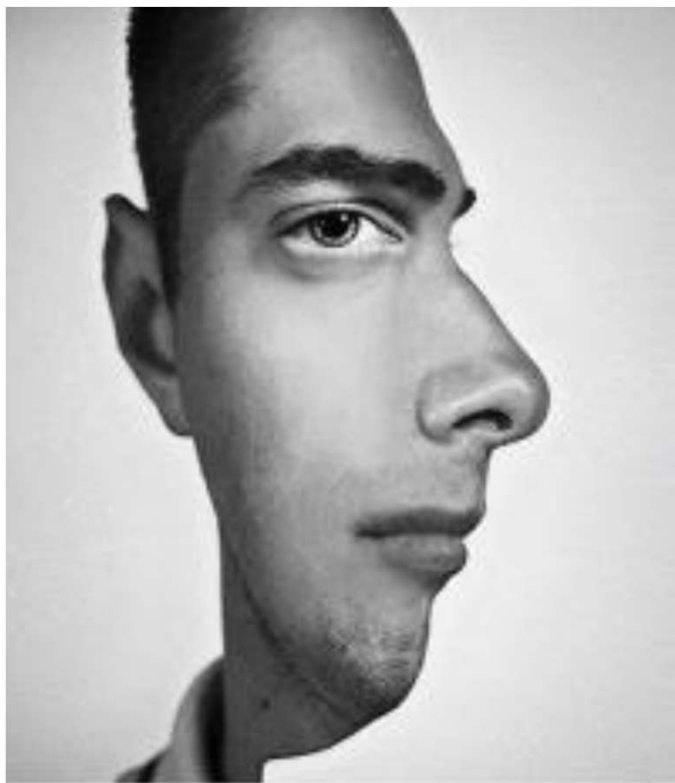


Figura 1 – Extração de *features* pelo cérebro irá definir o que o você vê.

Se olhar na esquerda da imagem irá ver o rosto frontal.

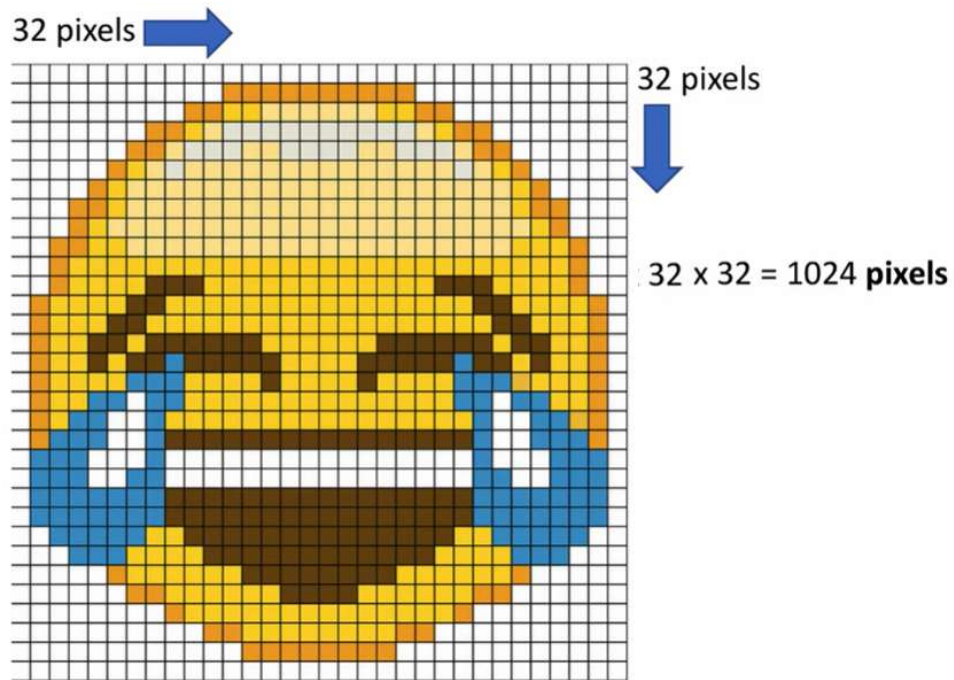
Se olhar no lado direito da imagem irá ver um homem de perfil.

Introdução a Redes Neurais Convolucionais

- Se as *features* extraídas não forem suficientes para classificação, iremos ficar confusos sobre a classificação do objeto.
- A ideia de redes neurais convolucionais é exatamente isso. Ao invés de entrar com uma imagem (pixel a pixel) numa rede neural convencional, conforme visto anteriormente, primeiramente vamos extrair *features* da imagem.

Pixels de uma imagem

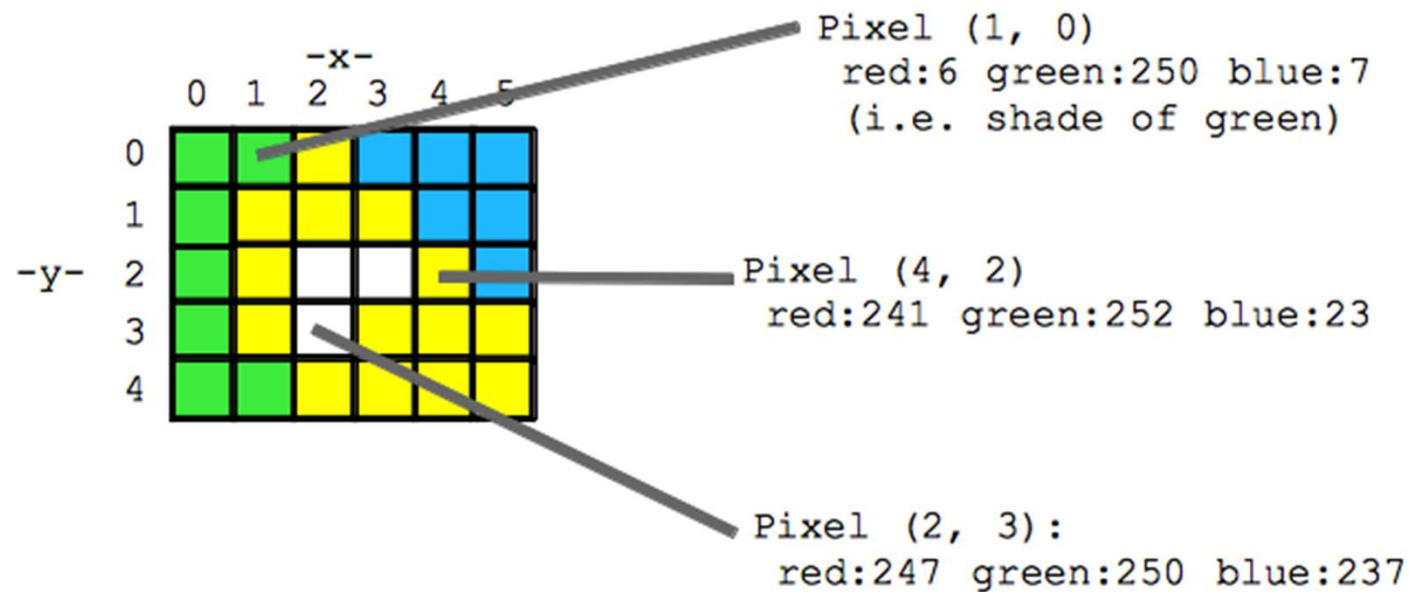
- Considere a figura abaixo de 32 x 32 pixels.



- Cada pixel é formado por uma combinação: RGB (Red, Green, Blue).

Pixels de uma imagem

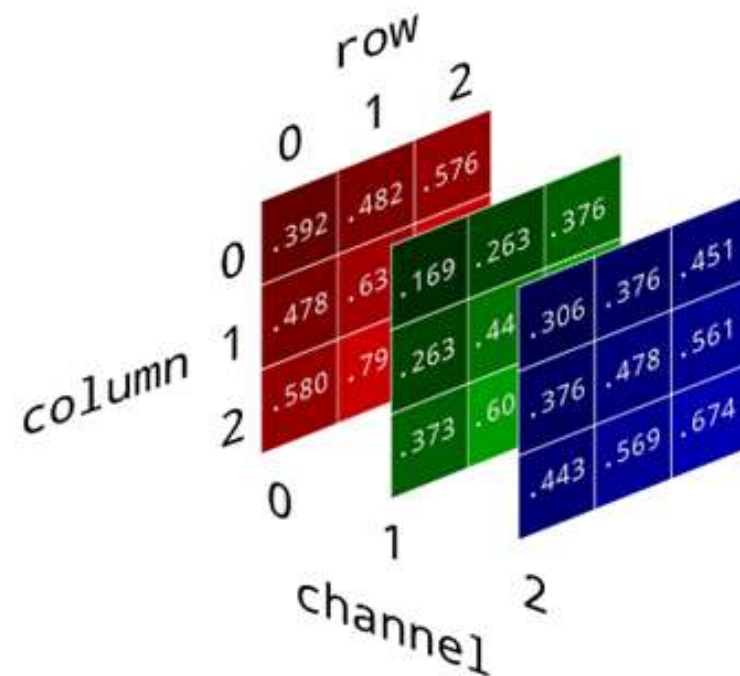
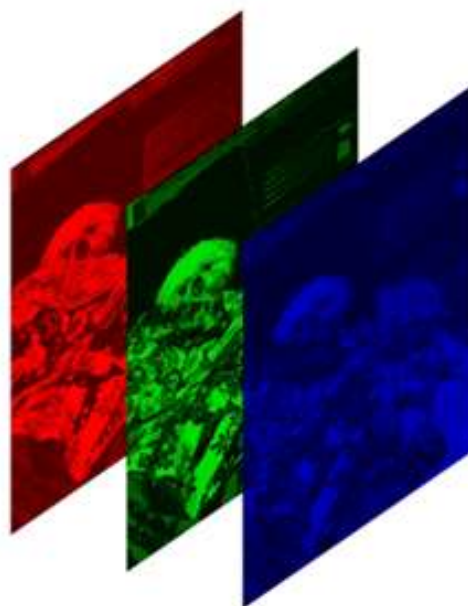
- Considere a figura abaixo de 5 x 5 pixels.



- Cada pixel é formado por uma combinação: RGB (Red, Green, Blue).

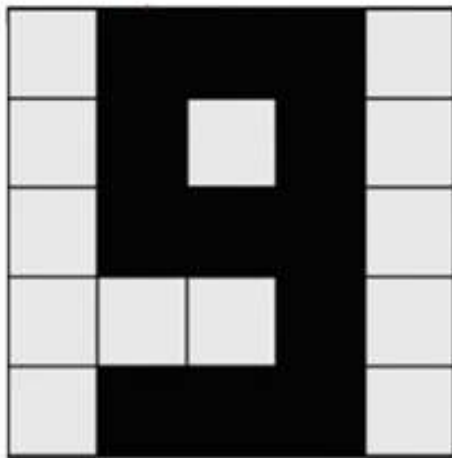
Pixels de uma imagem

RGB channels

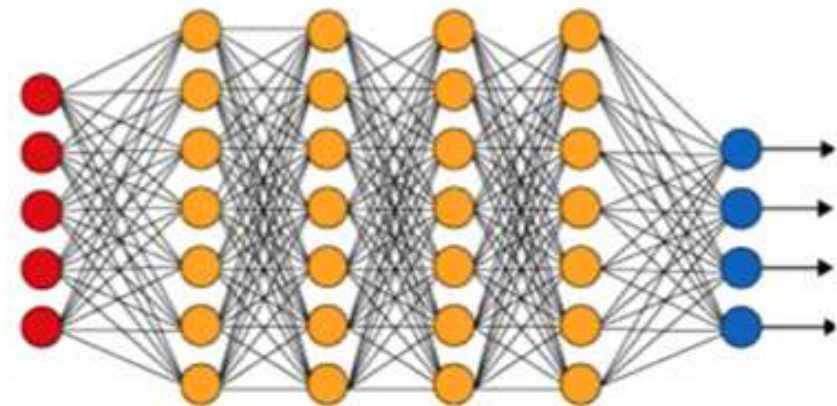


Pixels de uma imagem

- Considere a figura abaixo de 5 x 5 pixels (apenas 1 canal)
- Representação do pixel: 0: branco; 1: preto
- Passar estes pixels para a rede neural (25 neurônios na camada de entrada)



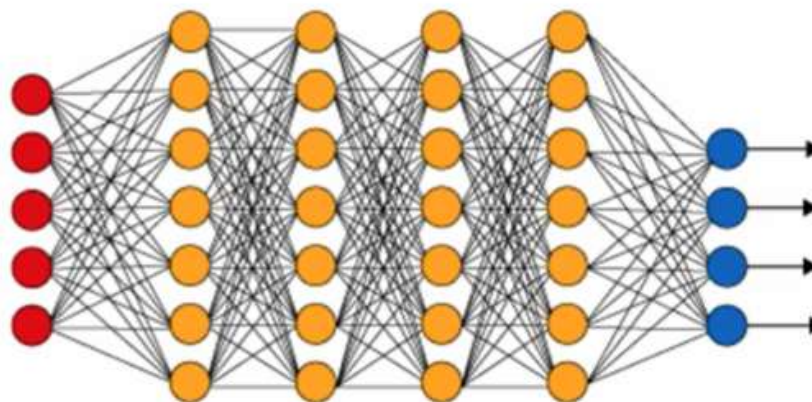
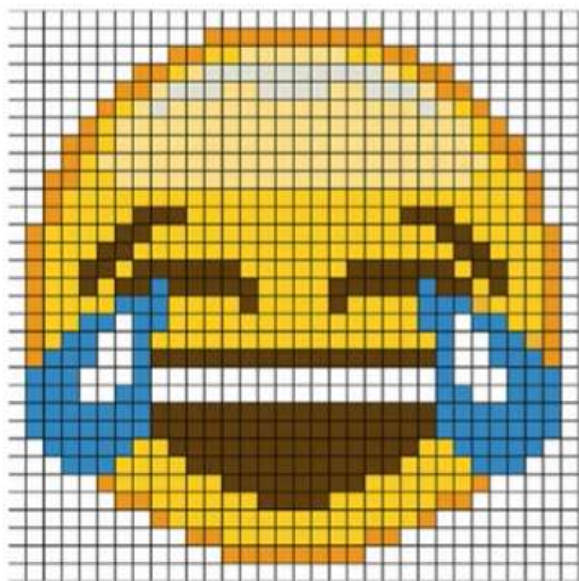
0,1,1,1,0,0,1,0,1,0,0,1,1,1,0,0,0,0,1,0,0,1,1,1,0,9



- Para uma imagem maior é complexo o processamento

Redes neurais densas x convolucionais

$32 \times 32 = 1.024 \times 3 = 3.072$ entradas



- Não usar todas as entradas (pixels)
- Usar uma rede neural tradicional, mas realizar uma transformação dos dados na camada de entrada
- Extrair as características mais importantes

Redes neurais convolucionais

- Etapas para construção da CNN
 - Etapa 1: Operador de convolução (aplicação de filtros)
 - Etapa 2: Pooling
 - Etapa 3: Flattening
 - Etapa 4: Rede neural densa

Etapa de Convolução

A extração de *features* leva em consideração a análise espacial da imagem através da convolução da imagem.

A convolução é o processo de adicionar cada elemento da imagem para seus vizinhos, ponderado por um kernel.

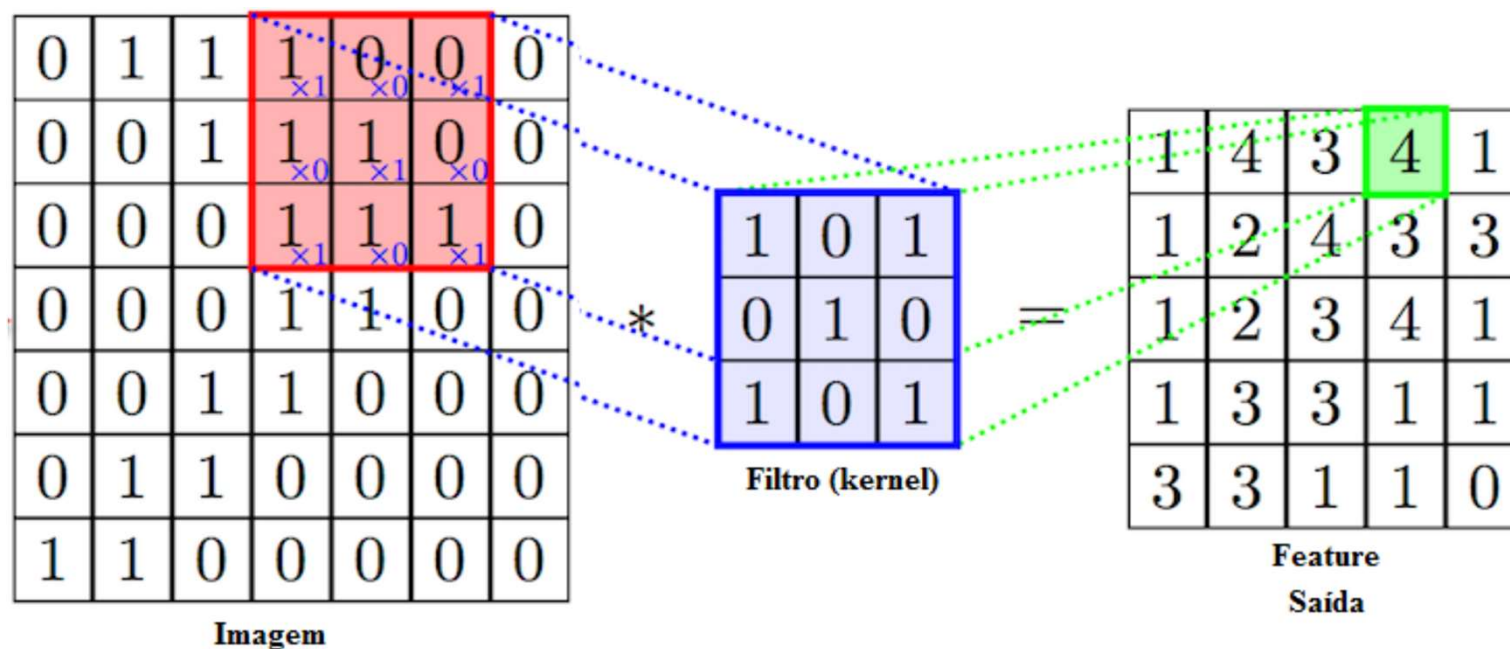


Figura 2 – Processo de convolução de um filtro que será passado na imagem e irá gerar uma nova matriz de saída.

Etapa de Convolução

- Na Figura 2 o filtro é passado por toda imagem. Na posição do filtro, haverá uma soma ponderada em todos os pixels. O resultado dessa soma irá ser o resultado no novo pixel na saída (verde).
- No exemplo da Figura 2 o filtro era uma matriz 3 x 3, no entanto, poderia ser qualquer outro valor, 9 x 9, 5 x 5, etc.
- O resultado da soma ponderada foi apenas uma saída (*stride*), no entanto, poderia ser outro valor, 2, 3, etc. O processo de passagem de diferentes tipos de filtros na imagem (*covolução*) é conhecido como mapeamento de features (*feature map*).
- Esse procedimento reduz o tamanho da imagem e extrai *features* das imagens, ou seja, detecta informações espaciais nas imagens.

Camadas de Convolução

- Na Figura 3 é apresentado uma imagem que representa o resultado de diferentes filtros passados pela imagem.

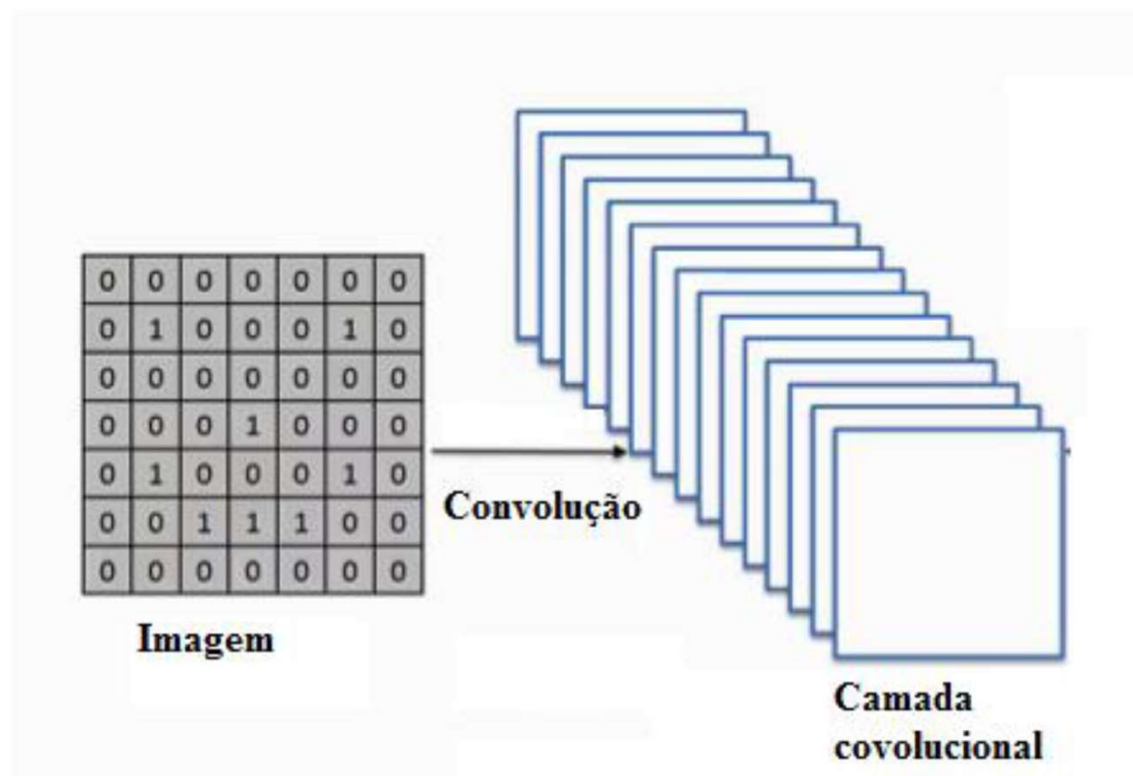


Figura 3 – Serie de camadas covolucionais após a passagem de diferentes filtros convolucionais na imagem de entrada.

Camadas de Convolução

- Na Figura 4 é apresentada uma imagem e o resultado após a passagem de diferentes filtros.

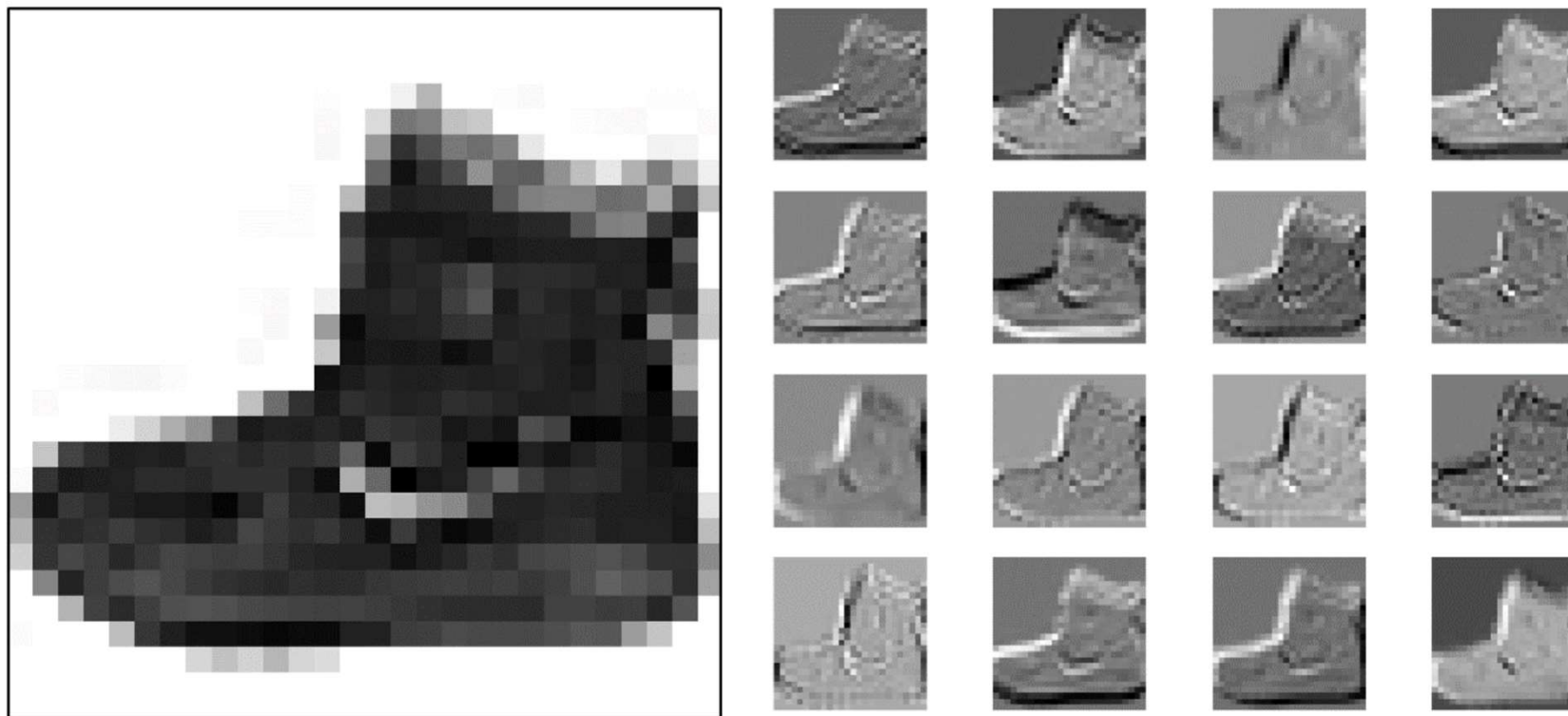


Figura 4 – Exemplo de uma imagem após a passagem de diferentes filtros

Camadas de Convolução

- Após da geração da camada covolucional, aplica-se a função retificadora na imagem para aumentar a não-linearidade (Figura 5). Então, esse processo é basicamente a aplicação da função “*relu*”.

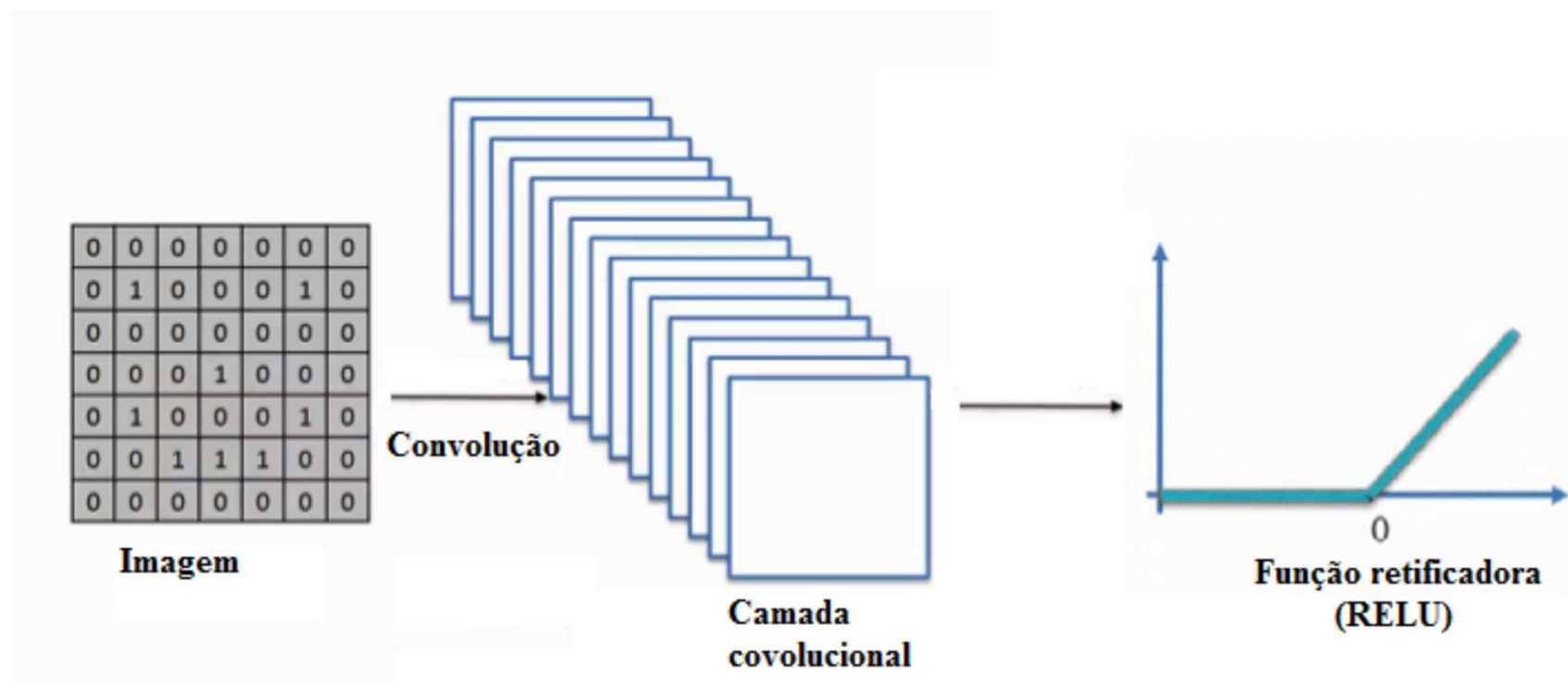
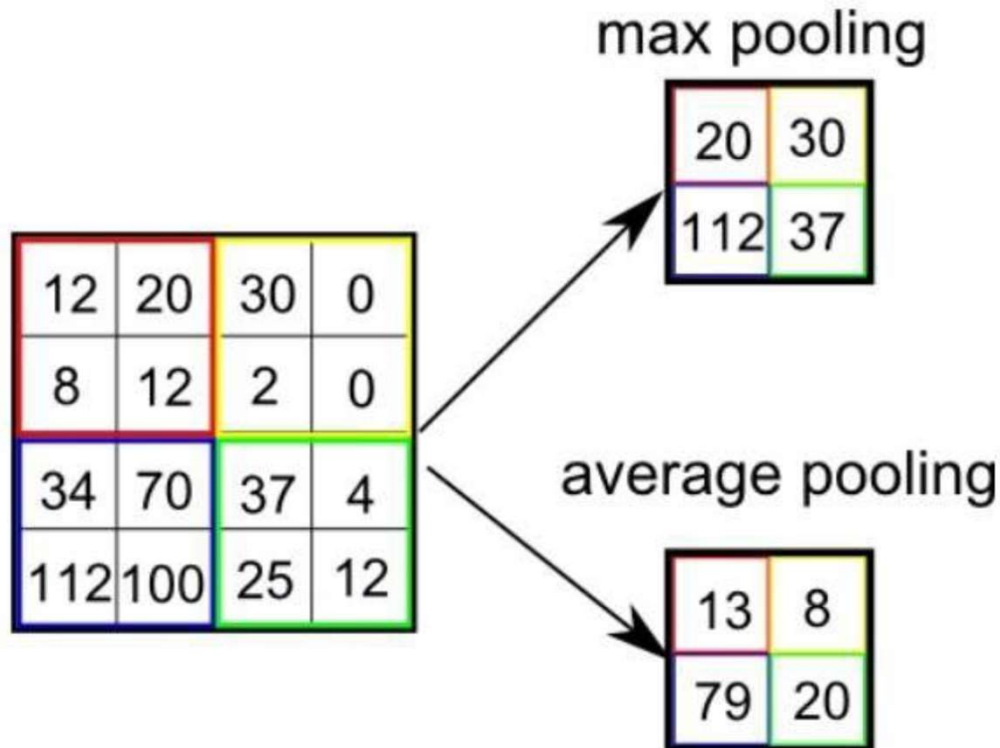


Figura 5 – Função retificadora aplicada na camada convolucional

Polling

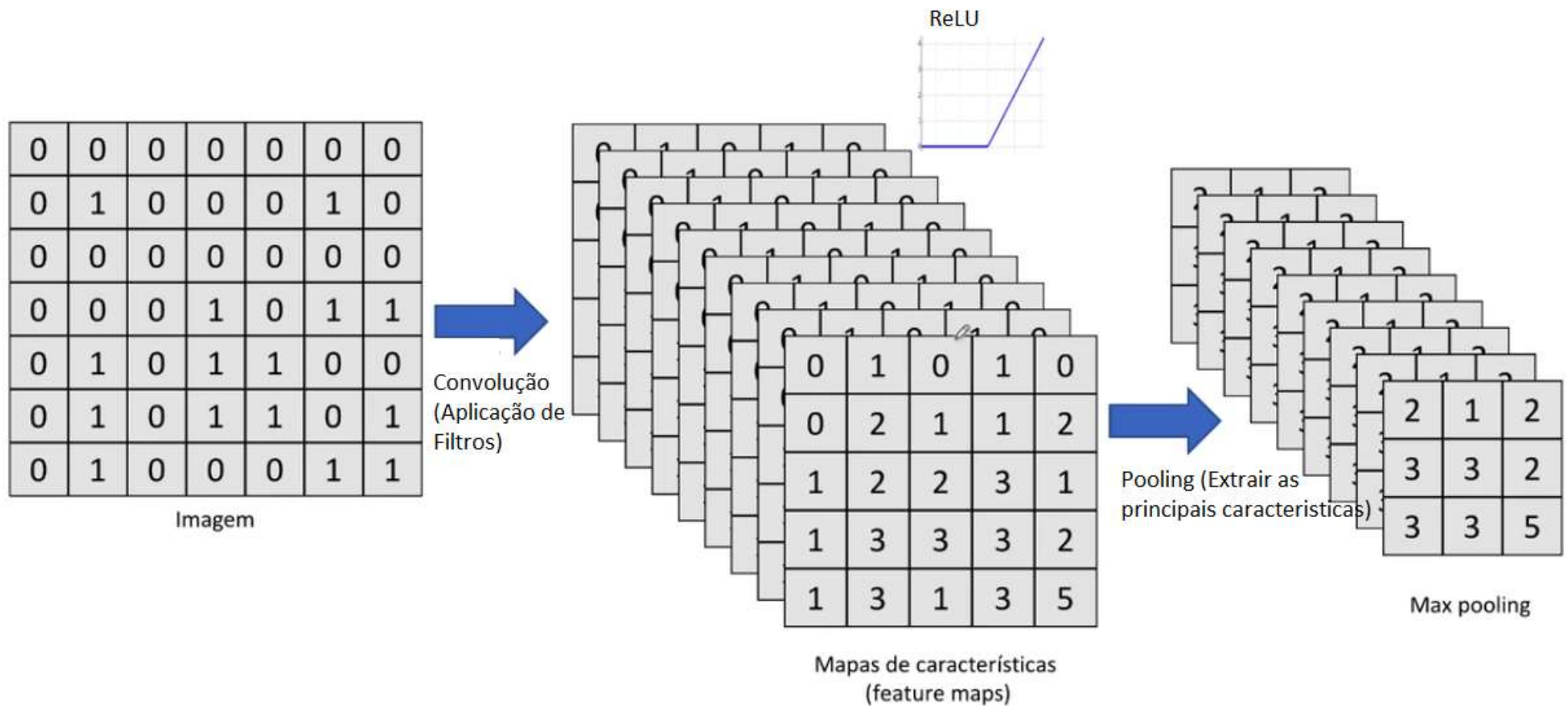
- Como uma imagem de um mesmo objeto pode estar em diferentes posições, rotações ou ângulos, então temos que aplicar uma nova etapa do processo para dar flexibilidade na localização das *features*, denominada *Pooling* (agrupamento de recursos).
- Então poderemos utilizar diferentes tipos de *Pooling*, *Max Pooling* (máximo do agrupamento), *Average Pooling* (média do agrupamento), etc.
- Na Figura abaixo é apresentado um exemplo da aplicação do *Max Pooling* e o *Average Pooling*.



Polling

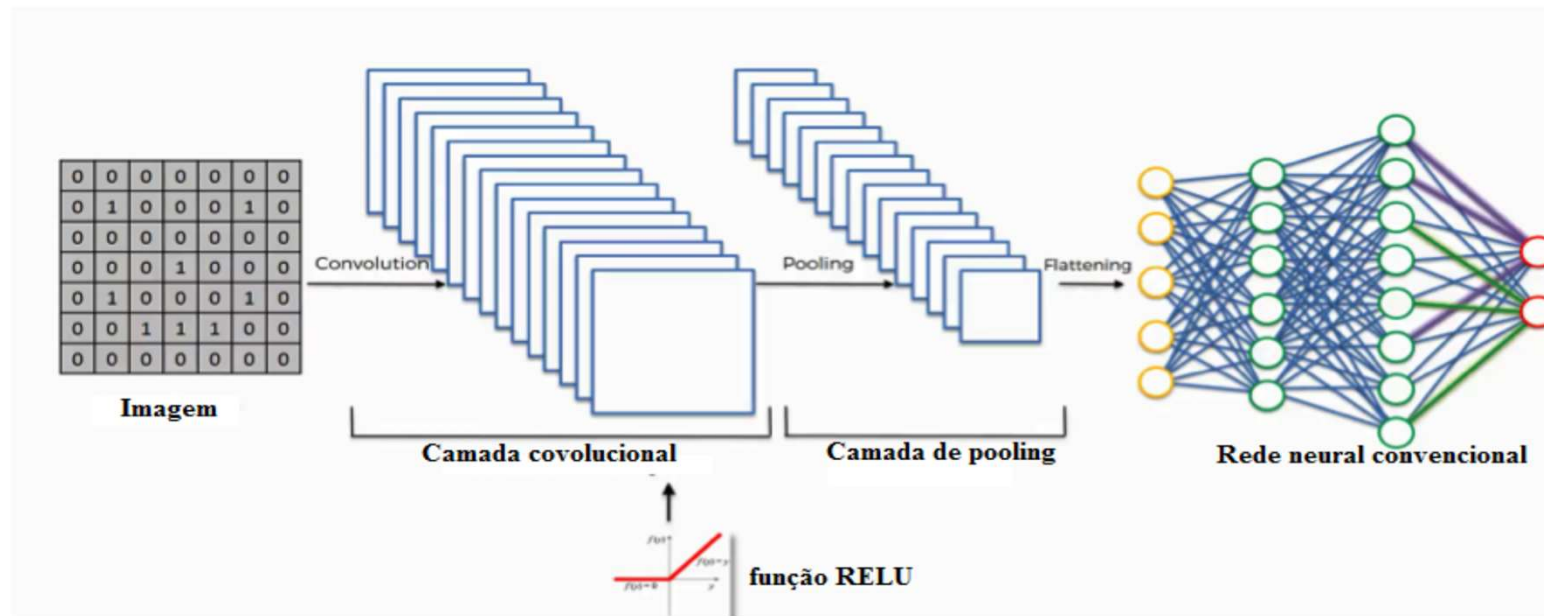
- Na Figura anterior é aplicado um filtro de 2 x 2 para aplicar o *pooling*.
- O *pooling* também reduz a quantidade de informações irrelevantes que irá entrar na rede neural final (full connection).
- Seleciona as características mais relevantes (reduz overfitting e ruídos desnecessários)
- Max polling: max foca nas características mais relevantes

Pooling



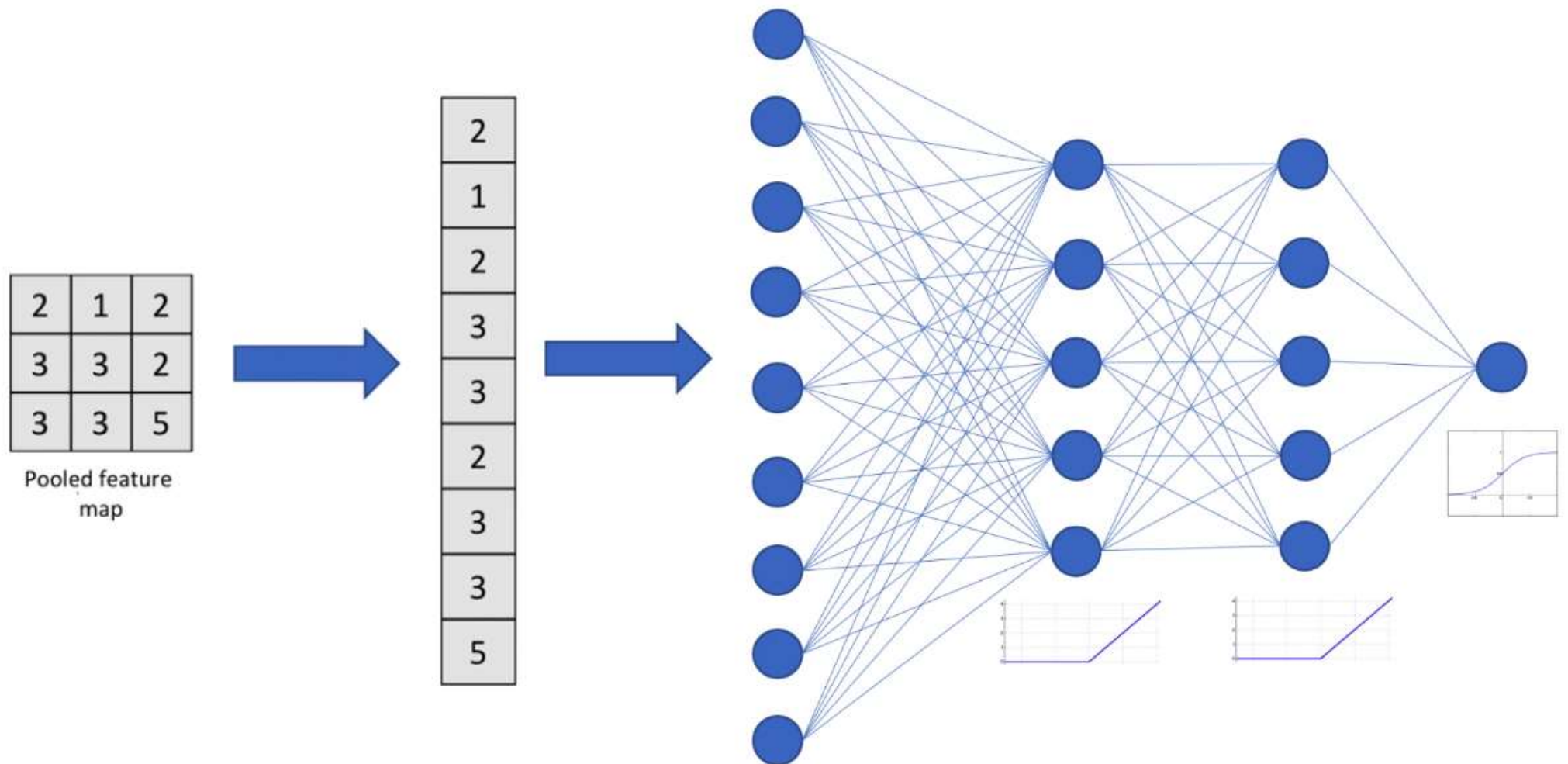
Flatten

- Agora temos que transformar a nossa matriz gerada pelo *pooling* em um vetor para entrar na rede neural artificial.
- Dessa forma será aplicado o *Flatten* na camada de *pooling*. Na sequência, após a geração do vetor, gera-se a rede neural convencional, conforme visto nas aulas anteriores.

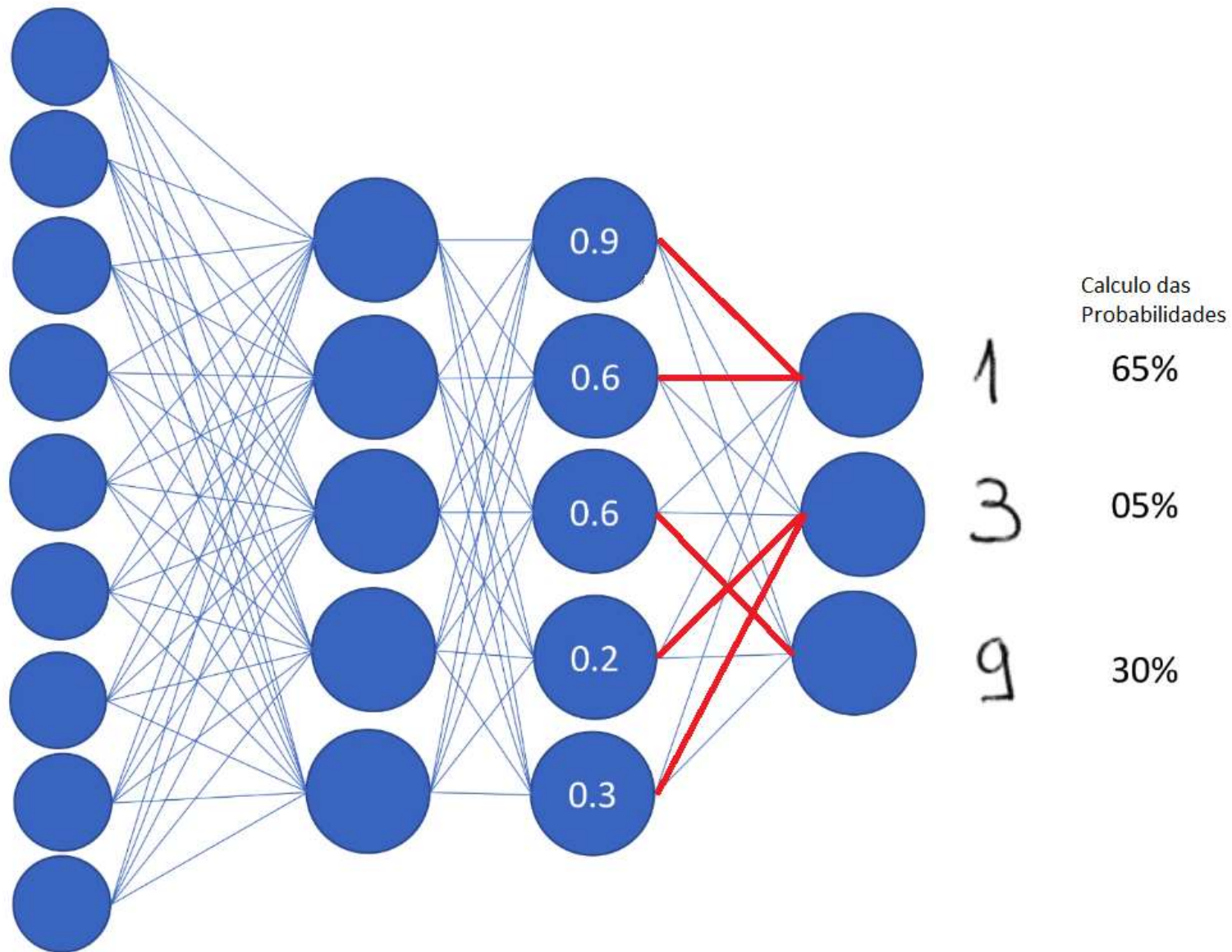


Na Figura acima é apresentado todo o processo da Rede Neural Convolucional.

Flatten



Rede Neural Densa Tradicional



Links Interessantes

- Demo: Convolução para reconhecimento de dígitos:
<https://deeplizard.com/resource/pavq7noze2>
- Convolutional neural networks – overview
<https://www.jeremyjordan.me/convolutional-neural-networks/>
- Introdução a Imagens Digitais
<https://web.stanford.edu/class/cs101/image-1-introduction.html>
- Explicação sobre kernels
[https://en.wikipedia.org/wiki/Kernel_\(image_processing\)](https://en.wikipedia.org/wiki/Kernel_(image_processing))
- Exemplo on-line.
<https://setosa.io/ev/image-kernels/>



INSTITUTO FEDERAL
Sudeste de Minas Gerais