

# 14 - Prática: Redes Neurais Convolucionais 1 (Deep Learning) (II)

- ▼ Entendimento rede neural convolucional

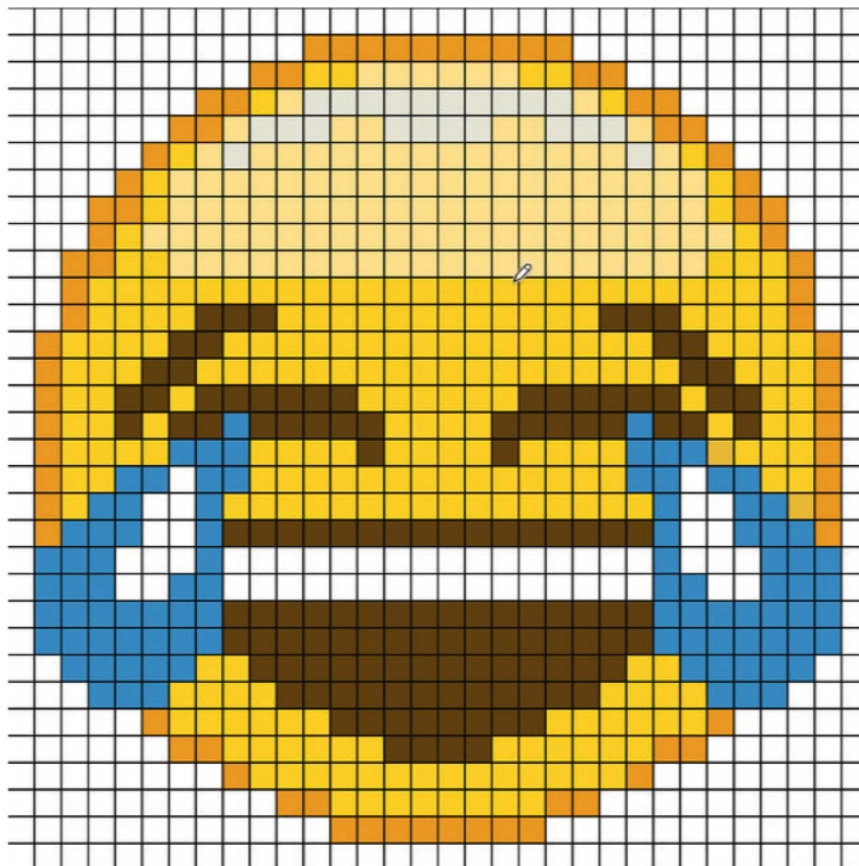
- ▼ Redes neurais convolucionais(CNN)

- Usado para visão computacional - Dar olhos ao computador, possibilitando ele enxergar

- Carros autônomos, detecção de pedestres(Uma das razões do Deep Learning funcionar bem)

- Em geral, melhor do que SVM(Support vector machines)

- ▼ Pixels



Combinação destes quadrados levam a criação de uma imagem.  $32 \times 32$   
= 1024 pixels

▼ Extração de características para identificação da imagem



Olho igual, cor igual, porém roupas diferem de cor, o que relacionado aos 2 pode ser considerada uma característica usual para

classificação.

#### ▼ Redes neurais densas X convolucionais

Redes convolucionais não utilizam todas as entradas (Pixels)

Utiliza uma rede neural tradicional, mas no começo transforma os dados na camada de entrada

Características mais importantes

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

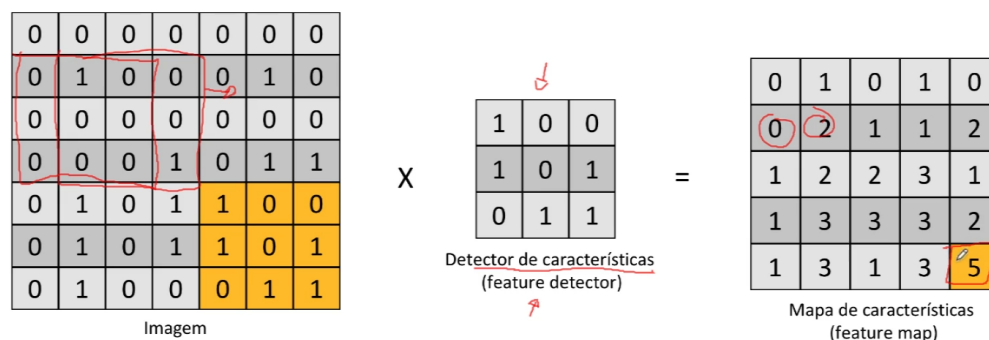
Imagem é uma matriz e o kernel outra matriz.

Logo é aplicado uma camada deste kernel para com a matriz original, realçando determinado atributo de uma imagem.

Após isto, temos os valores da imagem em questão, próximo passo é utilizar um detector de características para criação do mapa de característica.

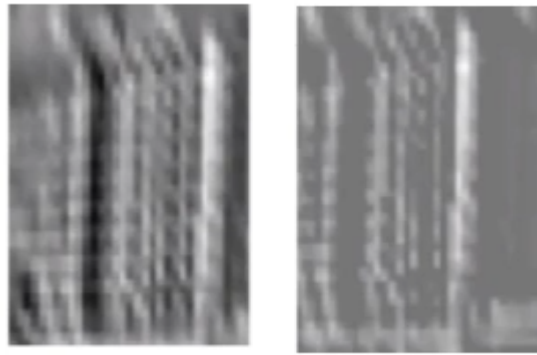
Imagem X Feature Detector = Feature Map

## Etapa 1 – Operador de convolução



Características podem ser perdidas, porém a ideia é identificar as partes principais, quanto maior for os números melhor será.

Por final, é necessária aplicação da função Relu, transformando valores negativos em 0 enquanto valores acima disso no que já estão.



Primeira imagem é sem utilização do Relu, enquanto segunda é com Relu. Ideia é padronizar as saídas em 0, afim melhorar a identificação.

- Camada de convolução - > Mapas de características. (Utilização de diversos mapas de características afim de testar e encontrar o que melhor se encaixa

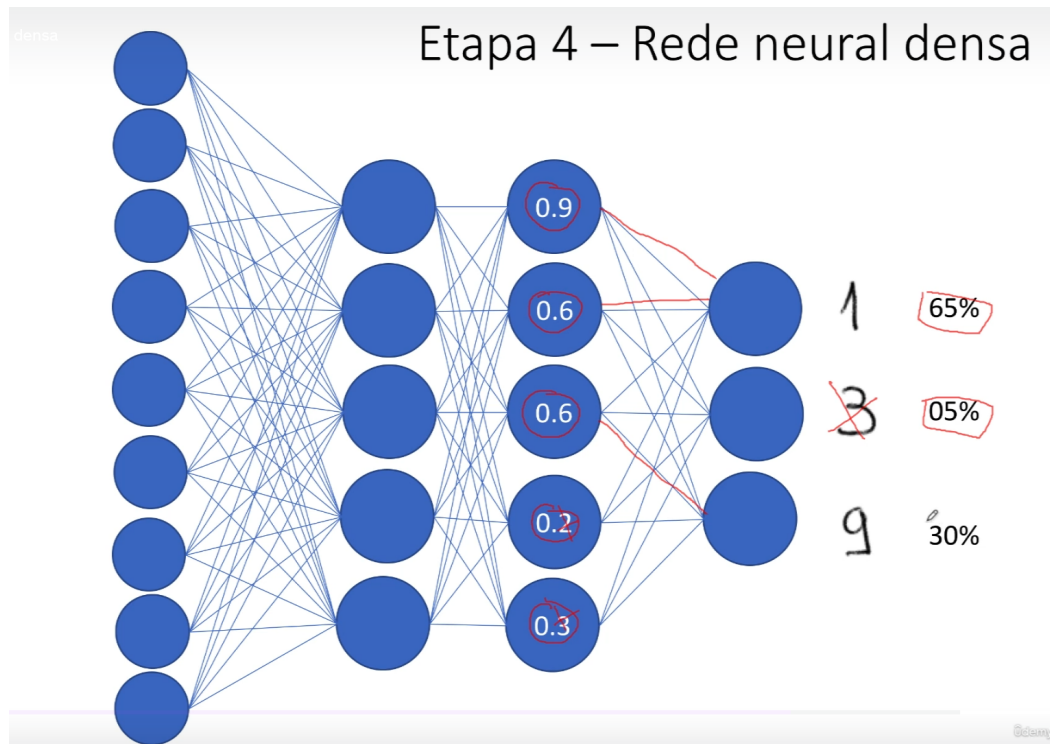
Pooling - Redução de overfitting e ruídos.

Max Poling(Mínimo, média): max foca nas características mais relevantes

Após ser realizado processo de polling é separado valores da matriz em um único vetor, do qual indica a quantidade dos nossos neurônios da nossa rede neural.

Rede neural Densa:

Pesos da rede neural: X valor de peso somados geram determinado número.



#### ▼ Base de dados MNIST

Código gerados a partir da base se encontram no github

#### ▼ Augumentation(Keras)

Processo de geração de uma imagem a partir de imagens já definidas no dataset.

Processo realiza alterações, que não comprometem diretamente uso da imagem, possibilitando mais testes com uma determinada classe

Na tentativa de utilização do augmentation, não foi possível rodar o pleno código, devido atualizações do TF.Keras quanto biblioteca.

```
AttributeError: 'Sequential' object has no attribute 'f
```

#### ▼ Resumo steps

1. Operador de convolução (Processa dados da entrada extraindo características relevantes)(Roupa Homer)
2. Polling(Reduzir a dimensionalidade dos dados, preservando características importantes)
3. Flattening(Tranforma uma matriz 2×2 em um vetor)

#### 4. Rede neural Densa(Camada de neurônios)