

# CNN

---

Advanced Institute for Artificial Intelligence – AI2

<https://advancedinstitute.ai>

## Agenda

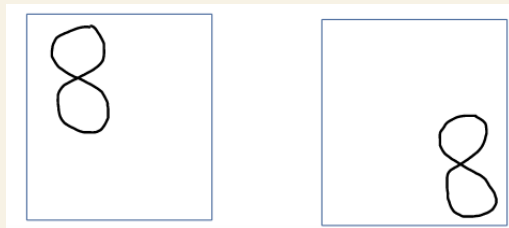
- ☐ Classificação de imagem
- ☐ Conceitos básicos CNN
- ☐ Arquiteturas CNN
- ☐ Reuso de arquiteturas (transfer learning)
- ☐ Pré-processamento
- ☐ Otimização do processo de treino

Aplicação de redes neurais para processamento de imagem:

- ☐ Classificação da imagem
  - Predizer as classes a qual uma imagem pertence
- ☐ Detecção de objetos
  - Classificação de imagem + identificar a localização de cada objeto presente
- ☐ Segmentação de objetos
  - Classificação de imagem + desenhar a borda de cada objeto presente na imagem

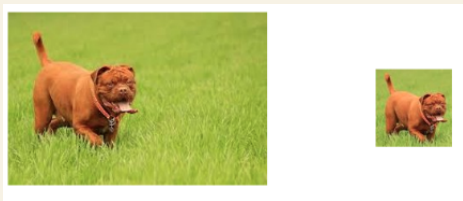
Desafio de classificar imagem:

- ☐ Distorções
- ☐ Diferenças de iluminação
- ☐ Translação



Para contornar tais problemas pode-se utilizar algumas estratégias

- ☐ Cortar e contralizar
- ☐ Separar o fundo da imagem
- ☐ Correções de iluminação, brilho, contraste, etc



Pre-processamento complexo e personalizado ao domínio

CNN possuem características que as tornam adequadas a atacar desafios relacionados a imagem:

- ☐ Sparse-connectivity: um único elemento no mapa de características é conectado a apenas um pequeno conjunto de pixels
- ☐ Compartilhamento de parâmetros: Os mesmos pesos são usados para diferentes regiões da imagem de entrada
- ☐ Muitas camadas: Combinando padrões locais extraídos com padrões globais

## CNN x MLP:

- ☐ Camadas densas aprendem padrões globais em seu espaço de features de entrada
- ☐ Camadas de convolução aprendem padrões locais: no caso de imagens, padrões encontrados em pequenas janelas da entrada Baseados em filtros (processamento de imagem)
- ☐ Os padrões aprendidos em uma convolução são invariantes à translação.
  - Os mesmos padrões podem aparecer várias vezes na mesma entrada em diferentes posições

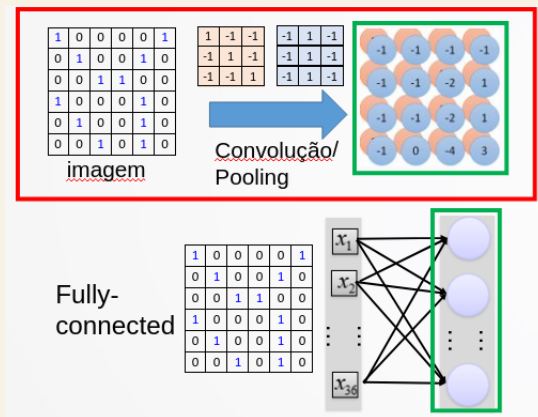
## CNN x MLP:

- ☐ Depois de aprender um determinado padrão no canto inferior direito da imagem, uma CNN pode reconhecê-lo em qualquer lugar: por exemplo, no canto superior esquerdo.
- ☐ Uma MLP precisaria de amostras para cada possibilidade de ocorrência do padrão na imagem
- ☐ CNNs precisam de menos amostras de treinamento para aprender representações, possuem maior poder de generalização no contexto de processamento de imagem



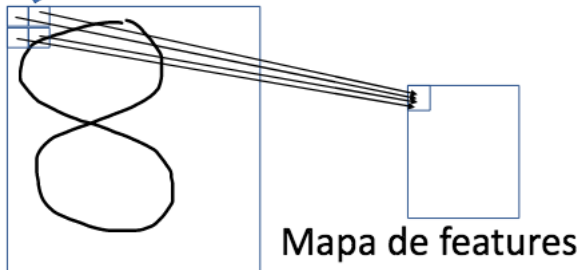
# CNN

CNN x MLP:

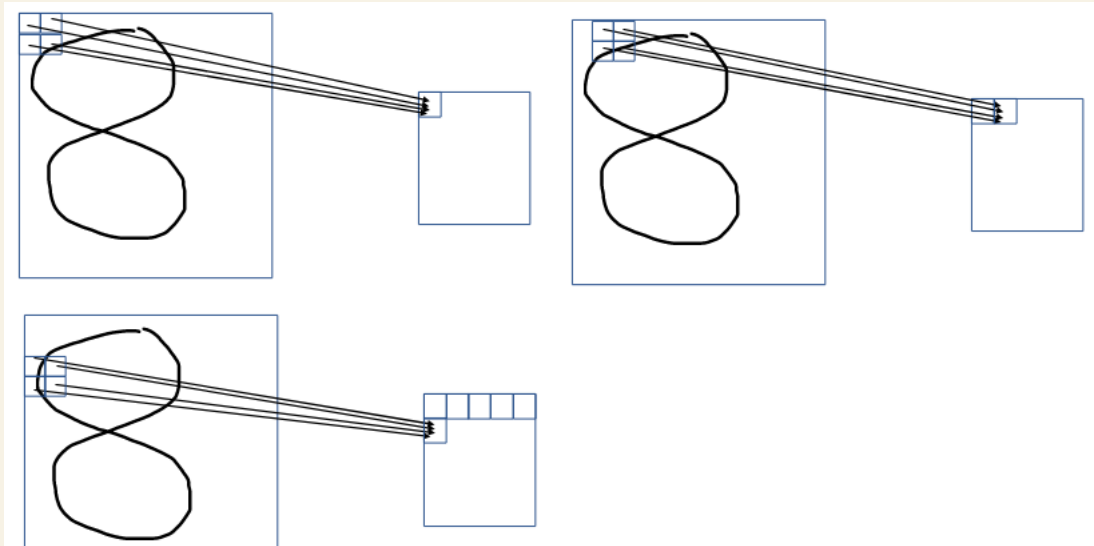


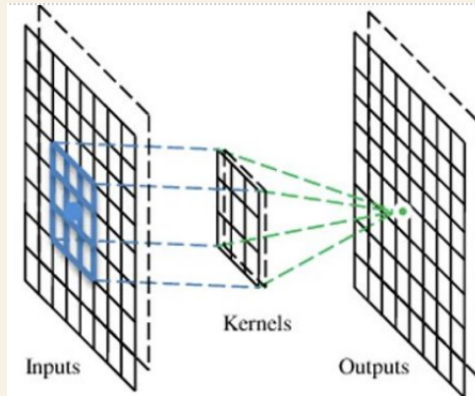
- ☐ Uma Matriz de Entrada (input), passa por um filtro (matriz Menor) e gera uma Matriz de Saída
- ☐ A aplicação desse filtro é chamado convolução
- ☐ Cada filtro é capaz de detectar um padrão em uma imagem

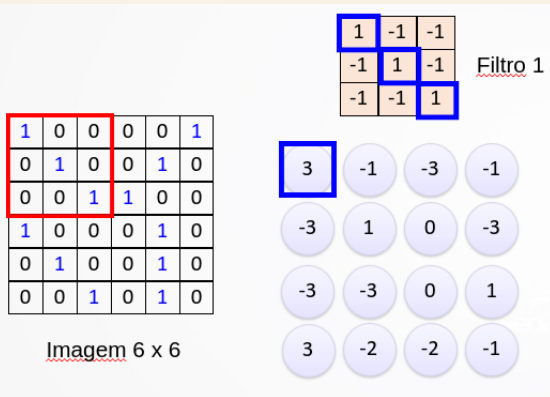
"feature detector" (filtro, kernel) desliza sobre as entradas para gerar um mapa de features



- Um detector de recursos que funciona bem em uma região também pode funcionar bem em outra região
- Grande redução de parâmetros para ajustar







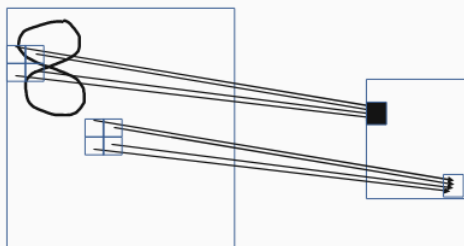
1	0	0	0	0	1
0	1	0	0	1	0
0	0	1	1	0	0
1	0	0	0	1	0
0	1	0	0	1	0
0	0	1	0	1	0

Imagem 6 x 6

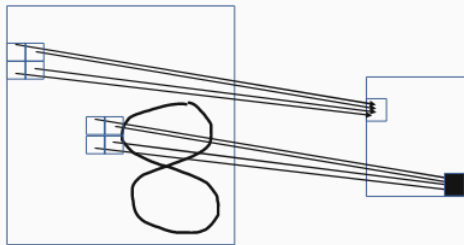
1	-1	-1
-1	1	-1
-1	-1	1

Filtro 1

3	-1	-3	-1
-3	1	0	-3
-3	-3	0	1
3	-2	-2	-1



**Invariante a escala, rotação e translação**

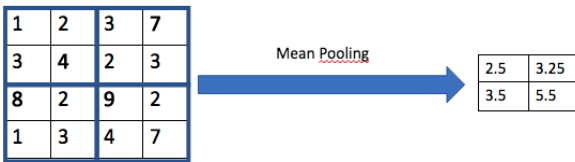
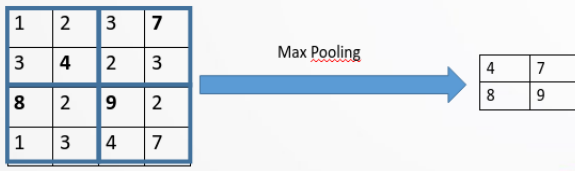


**Ativações são dependentes da localização**



As camadas do tipo pooling são importantes para generalizar a localização da ativação

- Dois métodos comuns de pooling:
  - pool médio : substituir todos os pixels de uma região pela média do valor dos pixels
  - pool máximo: substituir todos os pixels de uma região pelo pixel de maior do valor no intervalo
- A desvantagem é a perda de informação
- Tipicamente camadas de pooling não tem parâmetros a serem aprendidos



## Parametrização das camadas CNN

- ☐ Kernel: tamanho e formato da matriz de máscara
- ☐ Stride: passos percorridos a cada passada da máscara
- ☐ Padding: inclusão de bordas de tamanho configurável ao redor da imagem

Tais parametros controlam:

- ☐ Forma de aprendizagem de parâmetros
- ☐ Perda de informação
- ☐ Formato da matriz ao passar pelas camadas

Uma Rede Neural Convolutacional para classificação de imagem é montada tipicamente da seguinte forma:

- ☐ A entrada pode ser definida no formato 2D ou 3D
  - Altura, largura e um canal de cor (escala de cinza)
  - Altura, largura e canais de cor da imagem
- ☐ Sequencia alternada de uma camada de convolução seguida por uma camada pool
- ☐ Ao final uma camada densa para classificação

## Alguns projetos de redes neurais

- ☐ VGG
- ☐ Inception
- ☐ DenseNet
- ☐ ResNet
- ☐ mobilenet

Normalmente, essas redes podem ser utilizadas diretamente pelo Tensorflow e PyTorch

Pré-processamento de imagem para entrada em uma rede neural:

- ☐ Ler os arquivos de imagem
- ☐ Decodificar conteúdo de imagem para matriz de Pixel RGB
- ☐ Converter para tensores de ponto flutuante
- ☐ Modificar a escala de 0 a 255 para 0 a 1

Geração sintética de imagens a partir de imagens existentes (data augmentation)

- ☐ Permite aumentar o dataset de imagens, apenas fazendo algumas alterações nas imagens atuais
- ☐ Tal operação aumenta a capacidade de aprendizagem da rede, e auxilia na estabilidade do modelo

## Alguns parâmetros de data augmentation

- ☐ rotation\_range: valor em graus para rotacionar as imagens aleatoriamente
- ☐ width\_shift e height\_shift: define um intervalo aleatório para aplicar operações de translação
- ☐ shear\_range: aplica aleatoriamente transformações de cisalhamento
- ☐ zoom\_range: aplica zoom em regiões aleatórias da imagem
- ☐ fill\_mode: estratégia para preencher pixels novos que podem surgir após uma operação de rotação ou mudança de altura ou largura



De modo geral os passos de pré-processamento são realizados utilizando

- ☐ Bibliotecas de processamento de imagem como OpenCV
- ☐ Todos os arquivos de entrada são pré-processados e uma grande matriz é armazenada em memória
- ☐ Processo inviável para datasets com um número considerável de arquivos

## Leitura por batches

- ☐ permite realizar tais operações ao abrir os arquivos
- ☐ Abre apenas os arquivos que serão utilizados nos próximos batches
- ☐ Facilita utilizar diferentes amostras a cada treino (basta alterar o diretório com arquivos)

## Reuso de Redes Neurais Convolucionais (Transfer Learning)

- ☐ Redes Convolucionais tem foco em identificar pontos comuns nas imagens e gerar matrizes muito similares para imagens da mesma categoria
- ☐ Tal capacidade de generalizar tais identificações podem ser reutilizadas em diversos problemas
- ☐ Para isso podemos utilizar uma topologia de rede neural conhecida, e apenas alterar parte da rede
- ☐ Uma desvantagem de reutilizar uma topologia conhecida é a necessidade de retreinar a rede completa

## Reuso de Redes Neurais Convolucionais (Transfer Learning)

- ❑ Construir uma rede neural a partir dos pesos treinados de uma rede existente
  - Dessa forma, não é necessário re-escrever o código, da topologia conhecida
- ❑ Escolher camadas para não serem retreinadas
  - Esse processo otimiza o tempo de treinamento pois parte da rede permanece da forma original
- ❑ Incluir e retirar camadas ao modelo existente
  - Uma operação muito comum é alterar a última camada de uma rede convolucional de classificação
  - Dessa forma, o modelo utiliza toda a capacidade de uma rede conhecida, porém de forma especializada a um problema específico

Large Scale Visual Recognition Competition (ILSVRC) Diversas edições dessa competição foram realizadas com o intuito de avançar no desenvolvimento de redes neurais convolucionais que pudessem servir de base para resolver diversos problemas A seguir um gráfico com resultados de erros na classificação

