Advanced Institute for Artificial Intelligence

Processamento de Imagem

Advanced Institute for Artificial Intelligence – AI2

https://advancedinstitute.ai

Processamento de Imagem

Agenda

- Pré-Processamento
- Data Augmentation
- Reuso de topologias

CNN

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

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

CNN

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

CNN

Rotina de pré-processamento de imagens do Keras

- 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)

CNN

- Um recurso muito usado no pré-processamento de imagem é o data augmentation
- Tal recurso 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

CNN

Algumas opções para gerar novas imagens

- 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: splica aleatóriamente 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

Reuso de Redes Neurais Convolucionais

- 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

CNN

O Keras oferece recursos para treino de redes neurais que permite gerenciar diversos aspectos do processo de treinamento:

- Construir uma rede neural a partir dos pesos treinados de uma rede existe
  - Dessa forma, não é necessário re-escrever oo 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

CNN

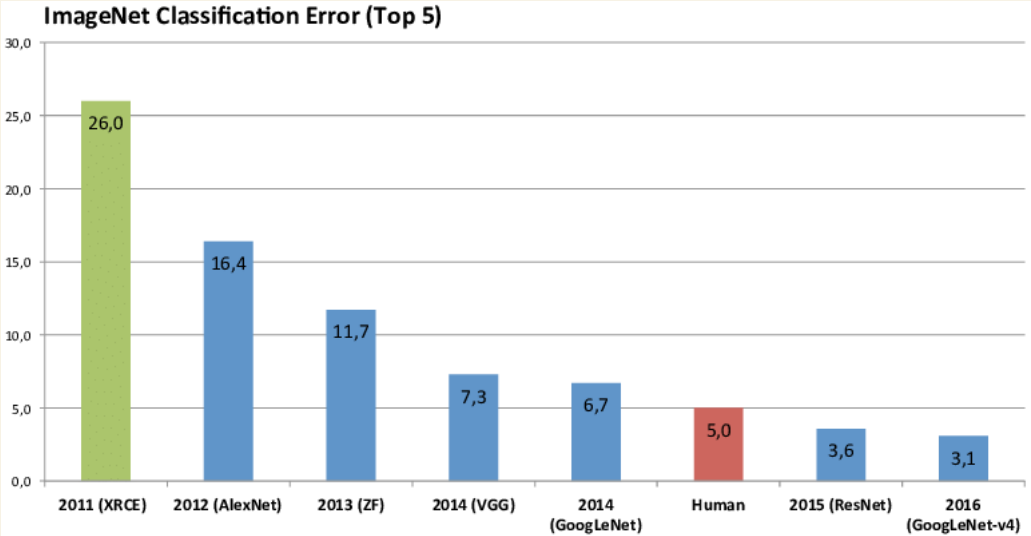
Muitas redes Neurais tem sido desenvolvida com foco em resolver problemas gerais:

- Classificação da imagem
  - predizer as classes de objetos presentes em uma imagem
- Localização de objeto único
  - classificação de imagem + desenhe uma caixa delimitadora em torno de um exemplo de cada objeto presente
- Detecção de objetos
  - Classificação de imagem + desenhe uma caixa delimitadora em torno de cada objeto presente

CNN

- Uma grande motivação para o surgimento dessas redes neurais é o ImageNet 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

CNN


ImageNet Classification Error (Top 5)
30,0
25,0
20,0
15,0
10,0
5,0
0,0
26,0
16,4
11,7
7,3
6,7
5,0
3,6
3,1
2011 (XRCE)
2012 (AlexNet)
2013 (ZF)
2014 (VGG)
2014 (GoogLeNet)
Human
2015 (ResNet)
2016 (GoogLeNet-v4)

CNN

Algumas Redes Convolucionais implementadas em Keras

- VGG
- Inception
- DenseNet
- ResNet
- mobilenet

CNN

Iniciando um modelo, marcando todas as camadas para não serem treinadas

vgg16 = VGG16(weights='imagenet', include_top=True)

for layer in vgg16.layers: layer.trainable = False

CNN

Utilizando slicing acessamos a penúltima camada (-2), e adicionamos uma nova camada como output da penúltime

x = Dense(1, activation='softmax', name='predictions')(vgg16.layers[-2].output)