

Machine Learning Engineer Nanodegree

Jean Shinohara

12/março/2018

Proposta

História e Justificativa

Classificação de imagens é um domínio dentro de Aprendizado de máquinas que possui inúmeras aplicações. Desde uma "simples" classificação entre cães e gatos, até diagnósticos de doenças ou rastreamento de pessoas as quais justificam a grande quantidade de pesquisas e desenvolvimento de modelos matemáticos e computacionais com o objetivo de disponibilizar soluções para essas diversas aplicações, sendo elas utilizando grandes recursos computacionais ou mesmo smartphones ou sistemas embarcados.

Dentre alguns modelos disponíveis, inclusive em algumas plataformas como Keras, podemos citar:

Modelo	Size	Top1 - Accuracy	Top5 Accuracy	Parameters	Depth
Xception	88MB	0.790	0.945	22,910,480	126
VGG16	528MB	0.715	0.901	138,357,544	23
VGG19	549MB	0.727	0.910	143,667,240	26
ResNet50	99MB	0.759	0.929	25,636,712	168
InceptionV3	92MB	0.788	0.944	23,851,784	159
InceptionResNetV2	215MB	0.804	0.953	55,873,736	572
MobileNet	17MB	0.665	0.871	4,253,864	88
DenseNet121	33MB	0.745	0.918	8,062,504	121
DenseNet169	57MB	0.759	0.928	14,307,880	169
DenseNet201	80MB	0.770	0.933	20,242,984	201

Ou seja, há diversos modelos disponíveis, mas qual é o mais adequado para cada uma das necessidades? Para responder a esta pergunta, irei propor um cenário simples de classificação de imagens, porém aplicável a uma solução com recursos computacionais limitados.

Descrição do Problema

As redes neurais convolucionais (CNN) tem se popularizado cada vez mais, principalmente depois da vitória no "ImageNet Challenge: ILSVRC 2012". Embora muitos modelos CNN estejam sendo desenvolvidos com o principal objetivo de aumentar sua precisão, vemos que a eficiência com respeito ao tamanho e velocidade não tem acompanhando na mesma velocidade. Isso traz implicações e limitações em sua aplicabilidade em soluções que necessitem uma resposta rápida e precisa como por exemplo, robótica, carros autônomos e realidades aumentadas.

Para atender a crescente demanda por soluções de Aprendizado de máquina e Inteligência Artificial e implementações em dispositivos Móveis, como smartphones e carros autônomos, a proposta deste projeto é o de avaliar o modelo MobileNet e aplicá-lo na classificação de imagens de cães e gatos, utilizando a base de imagens disponível no banco de dados da Kaggle.

O modelo MobileNets^[0] foi criado em um estudo de dispersão rígida para classificação de imagem^[1] e posteriormente em um estudo realizado pelo Google na normalização em lote^[2] com o objetivo de reduzir o custo computacional nas primeiras camadas e, pretendo com este projeto, avaliar sua eficácia e precisão para um problema bem conhecido e estudado que é o de classificação de cães e gatos.

Dados e entradas

As séries de dados foram obtidos da plataforma Kaggle^[3] e contém dois conjuntos de dados, treinamento e testes. O diretório de treinamento contém 12500 imagens de cães e 12500 imagens de gatos os quais foram rotulados de acordo enquanto que o conjunto de testes possui 12500 imagens e o nome dos arquivos foram rotulados com um ID numérico sem qualquer rótulo. Ambas os conjuntos de imagens representam situações cotidianas aos cães e gatos.

Descrição da Solução

O modelo escolhido para este projeto é o MobileNet o qual será submetido a variações de diferentes parâmetros com o objetivo de encontrarmos a melhor modelagem para resolver o problema de classificação de cães e gatos.

Modelo de referência (benchmark)

O modelo de referência deste projeto é o utilizado por Howard A. et all no estudo realizado em 2017 para demonstrar a eficiência do modelo MobileNets sobre diversas aplicações e casos de usos como detecção de objetos, atributos de faces e geo-localização de grande escala.

O resultado obtido neste estudo foi bastante expressivo comparando com alguns modelos populares como GoogleNet e VGG16 o qual obteve uma precisão muito próxima a estes modelos, porém com ganhos no tamanho e nos custos computacionais. Comparando com VGG16, o MobileNet teve um ganho de 32 vezes no tamanho e 27 vezes no custo computacional, enquanto que comparado com GoogleNet, ele obteve um ganho de 2.5 vez no custo computacional e tamanho.

Modelo	ImageNet Accuracy	Million Mult-Adds	Million Parameters
1.0 MobileNet-224	70.6%	569	4.2
GoogleNet	69.8%	1550	6.8
VGG 16	71.5%	15300	138

Métricas de avaliação

Como o objetivo deste projeto é avaliar a precisão e custo de processamento, as métricas estarão alinhadas a este objetivo como a precisão e o tempo de processamento em cada uma das etapas (treinamento, validação e testes).

Para isso, será considerado o modelo ResNet-50^[4] como referência.

Desenho do projeto

Uma vez definido o objetivo do projeto e qual modelo a ser aplicado e estudado. O projeto será executado seguindo o desenho abaixo:

1) Buscar e armazenar o conjunto de dados do Kaggle a ser utilizado no estudo. Os dados a serem utilizados no treinamento, validação e testes serão baixados da plataforma Kaggle^[3].

2) Processar as imagens

2.1) Dividir as imagens em conjuntos de treinamento e validação

As imagens disponíveis no diretório train, serão separadas em mytrain e myvalid

2.2) Ajustar o tamanho e escala das imagens

As imagens serão ajustadas para um tamanho de 224x224 e os valores de RGB deverão ser escalados para valores entre 0 e 1.

3) Construir a estrutura do modelo de classificação

A construção do modelo passará pela importação das bibliotecas requeridas, definição do modelo de camadas e ativações, podendo ser experimentadas diversas possibilidades para atingimento dos objetivos e sua compilação.

4) Executar o treinamento do modelo

Esta etapa utilizará a API do Keras para treinamento e validação do modelo o qual poderá ser reexecutado até atingirmos uma precisão de 90% e tempo de predição de 0.05s por imagem.

5) Aplicar o modelo no conjunto de dados de teste

Utilizando a API do Keras, aplicar o modelo treinado a um conjunto de imagens separadas para os testes. Essas imagens não serão utilizadas durante as etapas de aprendizagem e validação para evitarmos o overfitting do modelo.

6) Avaliar o resultado

Consiste na avaliação de todos os passos, uma vez que o projeto está interessado em avaliar o custo computacional das etapas de aprendizado, validação e testes, bem como avaliar a precisão do modelo.

7) Reportar o resultado

Criação do documento com os detalhes do projeto, execução e resultados no formato .PDF para submissão junto a Udacity.

Referências

[0] Howard A., Zhu M. et al. - MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications, Google Inc. arXiv:1704.04861v1 [cs.CV] 17 Apr 2017

[1] L. Sifre. Rigid-motion scattering for image classification. PhD thesis, Ph. D. thesis, 2014.

[2] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv preprint arXiv:1502.03167, 2015.

[3] <https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data>

[4] Liang Huiying, Capstone Project, December 7, 2016.