

Comparação de diferentes abordagens para resoluções do dataset cat's vs dog's

Comparing different approaches to solve Cat's vs Dog's dataset

Lucas Mrowskovsy Paim*

2019

Resumo

Este trabalho tem por objetivo comparar diferentes técnicas para a solução do dataset cat's vs dogs, a versão utilizada por este trabalho é uma base fornecida pelo Tensorflow com cerca de três mil imagens de gatos e cachorros para realizar a classificação de imagens, foram utilizadas as técnicas transfer learning utilizando modelo shallow e fine tuning e também criado uma rede neural densa a partir do zero. Todos os modelos tiveram resultados satisfatórios para o problema proposto.

Palavras-chave: deep-learning. tensorflow. cnn. cats vs dogs.

Abstract

This work aims to compare different techniques for the solution of the cat's vs dogs dataset, the version used by this work is a base provided by Tensorflow with about three thousand images of cats and dogs to perform the image classification, were used transfer learning techniques using shallow and fine tuning models and also created a dense neural network from scratch. All models had satisfactory results for the proposed problem.

Keywords: deep-learning. tensorflow. cnn. cats vs dogs.

*Programa de Pós Graduação em Inteligência Artificial, Curitiba - PR, 80215-901; Especializando em Inteligência Artificial Aplicada pela PUCPR; Tecnólogo em Sistemas para Internet pela Universidade Positivo; E-mail: lucasmpaim1@gmail.com

1 Introdução

Segundo [Géron \(2019\)](#), apesar de os computadores já terem conseguido feitos notáveis como vencer o campeão mundial de xadrez em 1996, eles ainda não conseguiam realizar tarefas triviais, como detectar um animal em uma foto, o que para nós humanos é algo extremamente trivial, isto ainda segundo [Géron \(2019\)](#) é devido ao nosso cérebro enviar para nosso subconsciente apenas informações de auto-nível, as redes neurais convolucionais (CNN), surgiram do estudo de nosso córtex visual e essas redes tem sido utilizadas como o principal meio para detecção de imagens desde então.

Neste trabalho será abordado algumas diferentes técnicas para a resolução do dataset cats vs dogs já pré-processada disponibilizada pelo Google na página do Tensorflow, onde foram removidas cerca de 1738 imagens corrompidas.

2 Método

O dataset utilizado por este trabalho, foi a "cats vs dogs" disponibilizada pelo Google em sua página da tensorflow, esta base contém 3 mil imagens de gatos e cachorros e seu propósito é ensinar a computadores a classificar fotos destes dois animais. A base está dividida em treinamento e validação, onde respectivamente temos, 2 mil imagens para treinamento e mil imagens para validação, ou seja, aproximadamente 30% da base voltada para validação.

As técnicas utilizadas para este trabalho são:

Transfer-learning baseada em extrator de características é utilizada uma rede previamente treinada como extrator de características, e estas são utilizadas para treinar modelos razos, como: SVM, KNN, etc.

Transfer-learning baseado em fine-tuning é utilizada uma rede previamente treinada, em que as camadas convolucionais do modelo são congeladas, e as camadas densas são retreinadas para o novo problema.

From scratch Criar e treinar um modelo a partir de um novo projeto, ou seja, sem reutilizar redes previamente treinadas.

2.1 Transfer Learning Shallow

O modelo pré-treinado utilizado por este trabalho é a Xception que é uma rede treinada utilizando a base [Imagenet \(2019\)](#), que é uma base de com milhares de imagens utilizada e alimentada pela comunidade científica.

Foram extraídos as características em arquivos CSV, esta extração resultou em 2048 características para cada uma das imagens, o dataframe de treinamento obteve o seguinte shape:

(2000, 2048)

Utilizando a técnica t-SNE, que segundo [Maaten \(2019\)](#), é uma técnica de redução de dimensionalidade estocástica, ou seja, utilizando esta técnica é possível reduzir um vetor de tamanho N para um de tamanho X , onde $X < N$, foi realizada a redução das bases de

características e apresentadas graficamente, para assim analisar o quão bem distribuídos estão as classes, estes gráficos podem ser conferidos na [Figura 1](#) e [Figura 2](#)

É possível notar estas características conseguem separar quase que perfeitamente as duas classes (gatos e cães), já na visão reduzida que temos na plotagem do t-SNE, com isso não é necessário um modelo muito robusto para que se obtenha resultados satisfatórios.

O modelo shallow escolhido para este trabalho é o Naive Bayes.

Segundo, [Britto \(2019\)](#), o teorema de Bayes consiste em uma abordagem probabilística para aprendizagem, calculando assim a probabilidade de uma característica X de um vetor de prever determinada classe.

A técnica de Naive Bayes, assume que todas as características no vetor são independentes, por isso, é chamado de Naive (do inglês Ingênuo), pois determinadas características podem estar relacionadas entre si, o que faz com que essa técnica tenha um resultado inferior a técnica de Redes Bayesianas que consideram que os atributos podem ou não ser relacionados, isto é, a característica X pode ou não estar diretamente relacionada a característica Y .

2.1.1 Resultados do Treinamento

Como esperado, o resultado foi extremamente satisfatório, conseguindo até 97,6% de acurácia.

Acurácia Naive Bayes Validação: 0.976

A matrix de confusão pode ser consultada na: [Figura 3](#).

2.2 Transfer Learning - Fine Tuning

Para o fine tuning deste trabalho, também foi utilizado a rede Xception, para uma melhor comparação em relação ao modelo shallow.

Para esta técnica além da rede xception que teve sua parte convolucional congelada, foi adicionada novas camadas para fosse realizado ajuste para o problema proposto, o sumário da rede proposta foi:

Model: "sequential"

Layer (type)	Output Shape	Param #
xception (Model)	(None, 2048)	20861480
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1024)	2098176
dropout (Dropout)	(None, 1024)	0
dense_1 (Dense)	(None, 2)	2050

Total params: 22,961,706
 Trainable params: 2,100,226
 Non-trainable params: 20,861,480

2.2.1 Resultados do Treinamento

Sua acurácia foi extremamente satisfatória tanto na base de validação quanto na base de teste, o que pode ser um indicador que a rede não terá um bom desempenho no processo de generalização:

loss: 0.0115 - accuracy: 0.9956 - val_loss: 0.0311 - val_accuracy: 0.9950

A matriz de confusão pode ser visualizada em: [Figura 4](#)

2.3 Model from Scratch

O modelo criado do zero, foi montado baseado no modelo proposto pelo [Tensorflow \(2020\)](#) e em algumas configurações da Xception, sua arquitetura proposta foi:

Layer (type)	Output Shape	Param #
conv2d_3 (Conv2D)	(None, 299, 299, 16)	448
max_pooling2d_3 (MaxPooling2)	(None, 149, 149, 16)	0
conv2d_4 (Conv2D)	(None, 149, 149, 32)	4640
max_pooling2d_4 (MaxPooling2)	(None, 74, 74, 32)	0
conv2d_5 (Conv2D)	(None, 74, 74, 64)	18496
max_pooling2d_5 (MaxPooling2)	(None, 37, 37, 64)	0
flatten_1 (Flatten)	(None, 87616)	0
dropout_1 (Dropout)	(None, 87616)	0
dense_2 (Dense)	(None, 512)	44859904
dense_3 (Dense)	(None, 1)	513
Total params:	44,884,001	
Trainable params:	44,884,001	
Non-trainable params:	0	

2.3.1 Resultados do Treinamento

O resultado preliminar foi de 85,1% de acurácia na base de teste e 68.8% na base de validação.

loss: 0.3271 - accuracy: 0.8510 - val_loss: 0.6158 - val_accuracy: 0.6886

É um resultado decepcionante considerando que um modelo shallow conseguiu 97% de acurácia, porém, não foram aplicados meios de se melhorar esta rede para esta quantidade de imagens, que seria aplicar técnicas de aumento de dados, ou seja, aplicar transformações na imagem para que cada imagem gere N outras imagens, o que ajuda no treinamento da rede.

A matrix de confusão pode ser consultada na: [Figura 5](#),

Referências

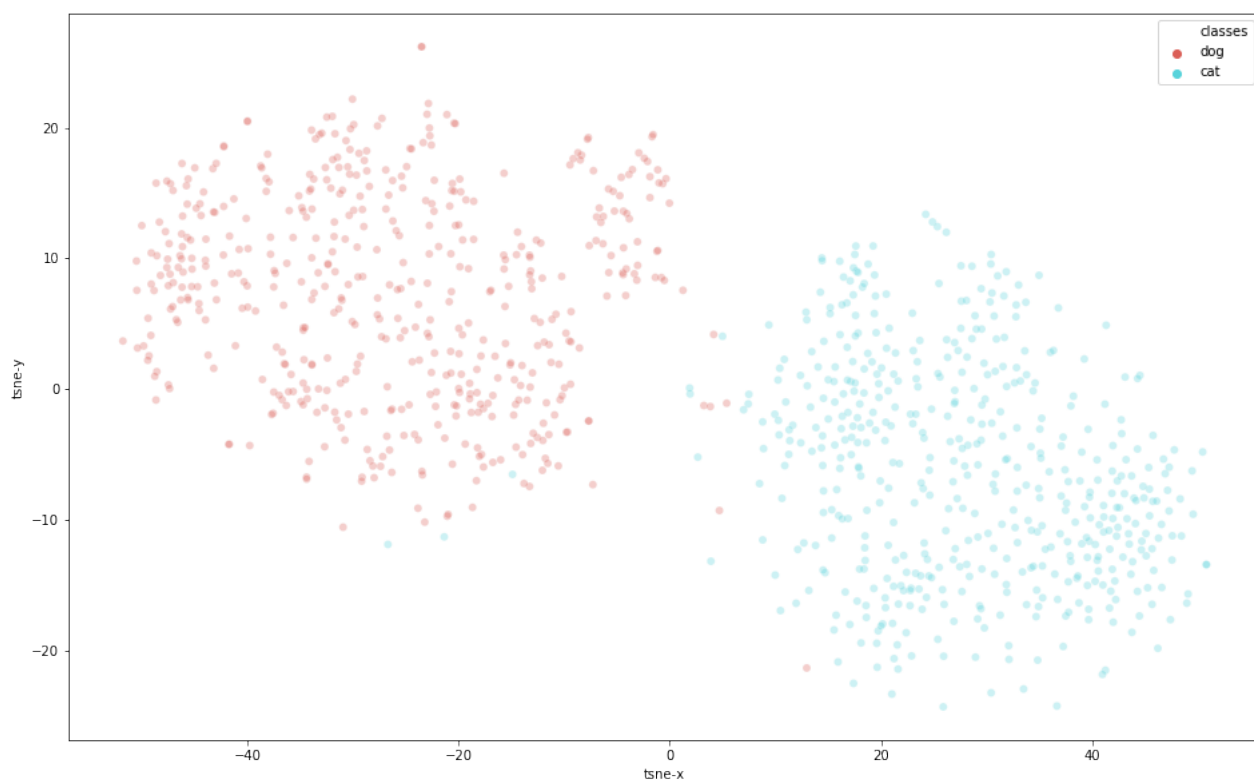
- BRITTO, A. *Aprendizagem Bayesiana*. 2019. Ppgia. Disponível em: <<http://www.ppgia.pucpr.br/~alceu/am/4%20-%20Naive%20Bayes%20and%20Bayesian%20Networks/NaiveBayes.pdf>>. Acesso em: 25 set 2019. Citado na página 3.
- GÉRON, A. *Mãos à obra: Aprendizagem de máquina com Scikit-Learn e Tensorflow*. 1. ed. Rio de Janeiro: Alta Books Editora, 2019. Acesso em: 21 ago 2013. Citado na página 2.
- IMAGENET. Imagenet. In: . [s.n.], 2019. Disponível em: <<http://www.image-net.org/about-overview>>. Acesso em: 16 feb 2020. Citado na página 2.
- MAATEN, L. van der. *Aprendizagem Bayesiana*. 2019. Lvdmaaten. Disponível em: <<https://lvdmaaten.github.io/tsne/>>. Acesso em: 25 set 2019. Citado na página 2.
- TENSORFLOW. In: . [s.n.], 2020. Disponível em: <<https://www.tensorflow.org/tutorials/images/classification>>. Acesso em: 18 feb 2020. Citado na página 4.

Figura 1 – TSNE - Treinamento - TransferLearning Shallow



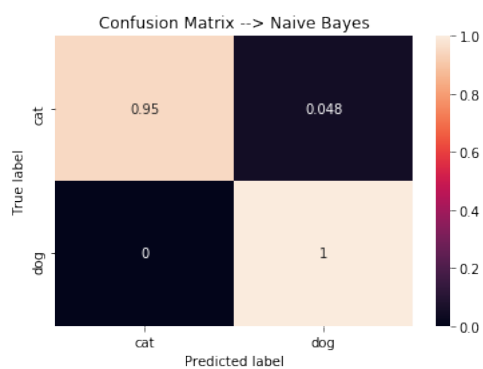
Fonte: O Autor

Figura 2 – TSNE - Validação - TransferLearning Shallow



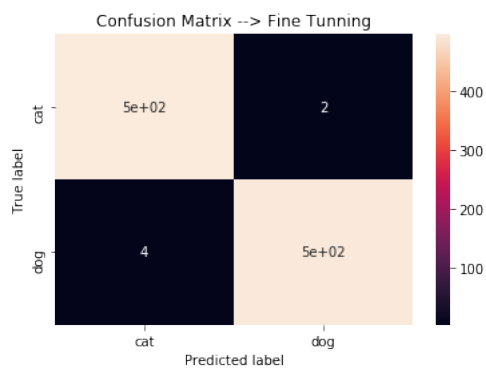
Fonte: O Autor

Figura 3 – Matrix de confusão - Naive Bayes - Normalizada



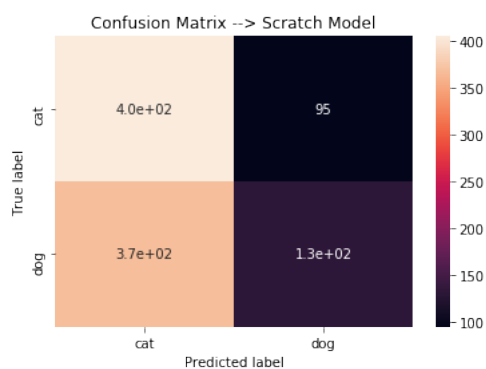
Fonte: O Autor

Figura 4 – Matrix de confusão - Fine Tunning



Fonte: O Autor

Figura 5 – Matrix de confusão - From Scratch Model



Fonte: O Autor