Machine Learning Engineer Nanodegree

Capstone Proposal

yfchiena
January 3, 2019

Proposal

Domain Background

Dogs vs. Cats is a competition topic for kaggle in 2013. The goal is to train a model to distinguish whether it is a cat or a dog from a given picture. This is a problem in the field of computer vision, and also a binary classifaction problem.

Deep learning is one of the most important breakthroughs in the field of artificial intelligence in the past decade. It has achieved great success in speech recognition, natural language processing, computer vision, image and video analysis, multimedia, and many other fields.

Hinton's research team proposed the AlexNet architecture and won the ImageNet Image Classification Competition in 2012, which the accuracy is more than 10% above the second place. Since then, the annual champions are all methods of neural networks.

Year	Model	Top 5 error rate
2012	AlexNet	15.4%
2013	ZFNet	11.2%
2014	GoogLeNet	6.7%
2015	ResNet	3.57%
2016	GBD-Net	2.99%
2017	SeNet	2.3%

Problem Statement

Input: A color photo.

Output: The probability(0~1) of a cat or a dog.

The project needs to identify cats and dogs, which is essentially a binary classification problem. Corresponding to supervised learning is to use the existing pictures with labels, and after the training is completed, the unlabeled pictures are classified. Therefore, it is also possible to solve this problem using a supervised learning method such as SVM. But due to the excellent performance of deep learning, I will choose the deep learning method to complete this project.

Datasets and Inputs

The datasets comes from Kaggle[1], there are 25,000 pictures in the training data, half each for cats

and dogs, and each picture has a category label. The test data has a total of 12,500 images. And the training:validation ratio is 4:1.

In all the above pictures, all the color pictures contain RGB three-channel information, but the picture quality are different, and the images size are inconsistent. There is no way to directly input it into the neural network, so 'resize' and 'pre-processing' are needed.

Solution Statement

Convolutional Neural Network (CNN) is one of the most representative network structures in deep learning technology. It has achieved great success in the field of image processing. In the international standard ImageNet datasets, many successful models are based on a CNN. One of the advantages of CNN over traditional image processing algorithms is that it avoids complex pre-processing of images (especially feature extraction) and can directly input the original image. The CNN performs multiple convolutional layers and pooling layers processing on the image, and in the output layer to obtain the respective probabilities of the two categories.

Benchmark Model

Choose keras-based network model-ResNet50 to complete the project. On kaggle, there are total of 1,314 teams participated in the competition. So we can compare our results with all teams to get a relative comparison.

In the Chinese version of the capstone project, there is a threshold for students to pass. The minimum requirement is reaching the top 10% (the score of 131 is 0.06127) of the kaggle Public Leaderboard. Maybe we can take this as the goal to get the score less than 0.06127.

Evaluation Metrics

A standard evaluation formula was proposed from Kaggle.

$$LogLoss = -\frac{1}{n}\sum_{i=1}^{n}[y_i log(\hat{y}_i) + (1-y_i) log(1-\hat{y}_i)]$$

where,

- n is the number of images in the test set
- $\hat{y}_i$ is the predicted probability of the image being a dog
- y_i is 1 if the image is a dog, 0 if cat
- $log()$ is the natural (base e) logarithm

'Log loss' is also called 'Logistic regression loss' or 'Cross-entropy loss', which is one of the commonly used evaluation methods in classfication problem.

Project Design

Datasets download and pre-processing

- Download images from Kaggle
- Split cats and dogs to be sorted in different folders
- Delete abnormal images
- Resize the image to keep the size consistent

Model construction and Feature extraction

- Use pre-train models from Keras
- Revise the fully connected layer
- Freeze all layers except fully connected layer to get the bottleneck feature

Model training and Result output

- Import trained weights to predict the result of testing set
- Ouput csv file and upload to Kaggle

Reference

[1] Kaggle Dogs vs. Cats datasets: https://www.kaggle.com/c/dogs-vs-cats-redux-kernels-edition/data