proposal

May 23, 2018

1 Machine Learning Engineer Nanodegree

1.1 Capstone Proposal

Bo Liu
May 22nd, 2018

1.2 Proposal

1.2.1 Domain Background

After centuries of intense whaling, recovering whale populations still have a hard time adapting to warming oceans and struggle to compete every day with the industrial fishing industry for food.

To aid whale conservation efforts, scientists use photo surveillance systems to monitor ocean activity. They use the shape of whales' tails and unique markings found in footage to identify what species of whale they're analyzing and meticulously log whale pod dynamics and movements. For the past 40 years, most of this work has been done manually by individual scientists, leaving a huge trove of data untapped and underutilized.

The challeng is to build an algorithm to identifying whale species in images. I will analyze Happy Whale's database of over 25,000 images, gathered from research institutions and public contributors. And I am excited to help open rich fields of understanding for marine mammal population dynamics around the globe.

Happy Whale is the organization who provid this data and problem. It is a platform that uses image process algorithms to let anyone to submit their whale photo and have it automatically identified.

1.2.2 Problem Statement

We have training data contains thousands of images of humpback whale flukes. Individual whales have been identified by researchers and given an Id. The problem we are going to solve is to predict the whale Id of images in the test set. What makes this such a challenge is that there are only a few examples for each of 3,000+ whale Ids. One relevant potential solution is training a model of CNN to identify thoes whale flukes.

Additionally, the problem can be quantified as follow, for an arbitrary image of whale flukes x (where x is the form of point series), and the corresponding whale Id y, find out a model $f(x)$ that can map x to a candidate Id $\hat{y}$.

The problem can be measured according to the precision at cutoff k.

The problem can be reproduced by checking the public datasets of whale flukes .

1.2.3 Datasets and Inputs

There are two floers contaning training images and testing images separately, and one csv file to map the training data to the right Id. - train - a folder containing 9850 training images of humpback whale flukes. - test - a folder containing 15610 test images to predict the whale Id. - train.csv - maps the training Image to the appropriate whale Id. Whales that are not predicted to have a label identified in the training data should be labeled as new_whale.

Each of the images including a humpback whale fluke, So appropriately identifying those image are related to solving the problem we described before.

The dataset can be obtained from kaggle website , or you can click here.

The training dataset will be used to train our model, while the test dataset will be used to test our model and compute the MAP@5.

For the reason that the total size of datasets is over 700MB, so I will not upload the datasets with my submition.

1.2.4 Solution Statement

A applicable solution to the problem is training a CNN model to predict the Ids.

The solution can be quantified as follow: Train a model $f(x)$ using training data x_{train}, applying the trained model $f(x)$ on the test data x_{test} to predict the condidate Id $\hat{y}$.

The solution can be measured by the precision at cutoff k.

The solution can be reproduced by reconstructe the same structure of CNN which are used in these project.

1.2.5 Benchmark Model

This link provided one possible solution using a traditional CNN in keras, which I would like to be taking into account as a benchmark model.

The workflow of the benchmark model is:

importing the data → One hot encoding on the labels → Image augmentation → Building and training model → Predictions on test samples

It is a simple CNN model containing 8 layers.

Input → conv2D@1 → Conv2D@2 → MaxPooling2D@3 → Conv2D@4 → MaxPooling2D@5 → Dropout → Flatten → Dense@6 → Dense@7 → Output@8

Where 'Input' represent input layer, 'conv2D' represent convolutional layer, 'MaxPooling2D' represent max pooling layer, 'Dense' represent fully-connected layer, 'Output' represent output layer, '@n' indicate that this is the 'n'th layer, 'dropout' and 'flatten' represent dropout and flatten method separately, and they aren't considered as a layer. All parameters shown below:

- batch size - 128
- epochs - 9
- conv2D@1 - number of kernals is 48, kernel size is (3, 3), strides is (1, 1), padding type is 'valid', activation function is 'relu'.
- conv2D@2 - number of kernals is 48, kernel size is (3, 3), strides is (1, 1), padding type is 'valid', activation function is 'sigmoid'.
- MaxPooling2D@3 - pool size is 3, 3), strides is (1, 1), padding type is 'valid'.

- conv2D@4 - number of kernals is 48, kernel size is (5, 5), strides is (1, 1), padding type is 'valid', activation function is 'sigmoid'.
- MaxPooling2D@5 - pool size is 3, 3), strides is (1, 1), padding type is 'valid'.
- dropout - Fraction of the input units to drop is 0.33.
- flatten - no parameters.
- Dense@6 - number of cells is 36, activation function is 'sigmoid'.
- dropout - Fraction of the input units to drop is 0.33.
- Dense@7 - number of cells is 36, activation function is 'sigmoid'.
- output@8 - number of cells is equal to the number of unique lables, activation function is 'softmax'.
- loss function - cross-entropy
- optimizer - Adadelta.
- metrics - 'accuracy'.

This model got an accuracy of xxx on training dataset, and xxx on testing dataset.
A script of this model is provided, name 'benchmark.py', in the submission package.

1.2.6 Evaluation Metrics

The solution can be measured by the Mean Average Precision @ 5 (MAP@5):

$$MAP@5 = \frac{1}{U}\sum_{u=1}^{U}\sum_{k=1}^{min(n,5)} P(k)$$

where U is the number of images, $P(k)$ is the precision at cutoff k, and n is the number predictions per image.

1.2.7 Project Design

A workflow for approaching a solution is described below:

acquiring data $\rightarrow$ exploring data $\rightarrow$ preprocessing $\rightarrow$ building model $\rightarrow$ training model $\rightarrow$ predicting on test data

loading data - Downloading and reading data with the right format.

exploring data - Exploring data using statistical method like the quantity of samples, depulicated images, etc.

preprocessing - Some preprocessing shold be done before diving into building a model, like one-hot encoding, converting image to gray, data augmentation, etc.

building model - Building a CNN model in tensorflow to solve this problem. Structure of the model will base on the classic CNN model called resnet.

training model - In this part, training data will be used to train the model that built in last step and I will use the MAP@5 evaluation metrics which is described in the Evaluation Metrics part to improve the model.

predicting on test data - This last step is to predict the whale Id using the trained model on testing data.