Machine Learning Engineer Nanodegree

Capstone Proposal

Ruba Rammal

March 9th, 2018

Domain Background

The Right Whales are species of whales that are threatened with extinction, so in order to protect the whales and monitor their health, researchers track their locations by taking photographs during aerial surveys. These photos are then matched by scientists against an existing catalog to determine the species by the aid of a software.

The system does not automate the process and depends on manual inspection which is time consuming and assumes that the researcher have knowledge in matching the whales so MathWorks sponsored a Kaggle[1] competition to automate this process using the dataset labeled by Christin Khan and Leah Crowe from NOAA and the photo-identification catalog provided by New England Aquarium.

Deep learning and convolutional neural networks made it possible to understand images and detect or recognize certain shapes and objects and make reliable predictions[2]. CNNs are suitable for automating whale recognition process

Problem Statement

To automate the process of identifying right whales which is an image classification problem, the labeled dataset that contains 447 unique whale ids will be used to build a model that can detect and classify right whales which will make the process of targeting the whale to monitor its health and help save it from extinction much more efficient.

The model will consist of a CNN that will classify to which class out of the 447 classes does the right whale in the input image most likely belong to.

Datasets and Inputs

The dataset compromises of different aerial images of 447 unique whales, the pictures were taken over the course of ten years using different equipment so the color of each image differs based on the time and equipment used to capture the photo, and each image contains a single whale. The images were picked and labeled by scientist at NOAA.

Each whale has a unique mark on its head that distinguishes it from other whales and this mark is not very clear in every image and that creates the challenge of identifying the whale. The images can be cropped prior to training the model to focus only on the whale and remove the sea water areas in the image.

The data is already split into training and testing sets and the training set contains 4544 images that are not well distributed, some whales have more images than others and many of the whales have only one image and that creates a challenge when training the model. The images are 2D and composed of three arrays for different color channels (RGB) but they vary in brightness and they range in dimensions as well since different equipment were used over ten years to capture them.

Solution Statement

Using the right whale images dataset, a convolutional neural network will be trained to give the probabilities of how likely does a right whale in the input image belong to each of the 447 classes. The model will take one image as an input and will output 447 probabilities using the log loss formula as required in the Kaggle competition description.

Benchmark Model

Using a one layer CNN architecture would yield a result that is slightly better than random chance which is no more than 0.9. The goal is to produce better result than that of a simple one convolutional layer CNN architecture or a simple regression model.

Evaluation Metrics

To evaluate the model, the log loss (or loss-entropy) formula will be used which produces a prediction values between 0 and 1.

$$Logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij})$$

The goal is to minimize this value so the closer the log loss value is to zero the better the prediction is.

Project Design

Software and Hardware

The model will be built using Python, Keras, scikit-learn, Pandas, NumPy and it will be trained on an AMD Radeon R9 M390 2048 MB GPU.

Data Pre-processing

To preprocess the images and focus on the head to tail part for training, the function provided in this Kaggle discussion post[3] will be used. Another way is by using OpenCV but that might be inefficient since the whales are not always centered in the image. One suction which was applied by the winner of the competition was to train a CNN to locate the head, align it and then crop the image[4]. According to previous

Convolutional Neural Network Design

A ten layer CNN architecture will be designed to build this model, it will include 3x3 filter convolutional layers with ReLU activation function, pooling layers and a fully connected layer with softmax activation function. The convolutional layers will detect shapes and edges from the

input images, the pooling layer will reduce the dimensionality of the features and the fully connected layer will produce the 447 outputs and it is fully connected with all of the nodes with the layer that precedes it. There is no need for a dropout layer since overfitting is not an issue in this domain. Transfer learning might be applied to this model to try to speed up the training time (using VGG19 or ResNet). Transfer learning means adding an existing pertained model for a problem to another similar problem. To tune the model, it is important to optimize the hyperparameters like wight initialization, learning rates and number of epoch. It is also possible to apply Keras's GridSearch to optimize the hyperparameters[5].
The CNN architecture is demonstrated in the following figure:

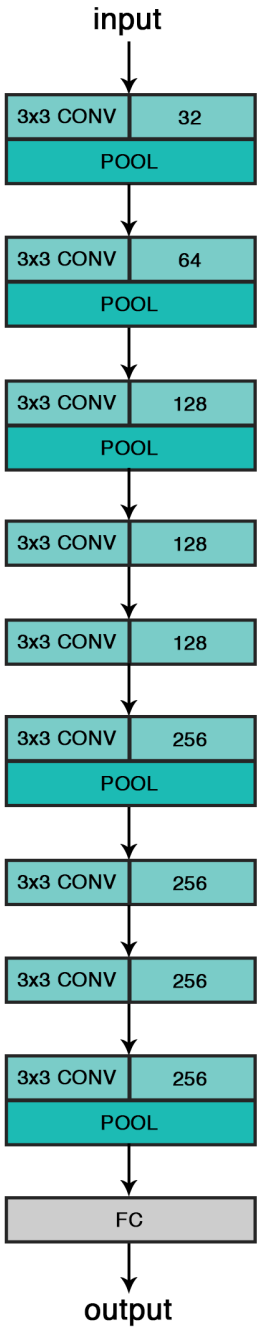

FIGURE 1 CNN ARCHITECTURE

References

[1] Kaggle, Right Whale Recognition, https://www.kaggle.com/c/noaa-right-whale-recognition
[2] Yann LeCun, Yoshua Bengio, Geoffrey Hinton. Deep learning, 28 May 2015, https://www.cs.toronto.edu/~hinton/absps/NatureDeepReview.pdf
[3] Kaggle, Right Whale Recognition, https://www.kaggle.com/c/noaa-right-whale-recognition/discussion/18251
[4] Robert Bogucki, Which whale is it, anyway? Face recognition for right whales using deep learning, 16 January 2016, https://blog.deepsense.ai/deep-learning-right-whale-recognition-kaggle/
[5] Jason Brownlee, 9 Agudt 2016, How to Grid Search Hyperparameters for Deep Learning Models in Python With Keras, https://machinelearningmastery.com/grid-search-hyperparameters-deep-learning-models-python-keras/