Machine Learning Engineer Nanodegree

Capstone Proposal

Isabel María Villalba Jiménez
Saturday 22nd October, 2016

Proposal

Right whales are one of the most endangered species around the world, with only a few 400 remaining. Many of casualties among them are caused by crashing into boats. One way of avoiding these collisions is to alert ships when whales are detected in the proximity.

In order to do so, Cornell University's Bioacoustic Research Program, which has extensive experience in identifying endangered whale species, has deployed a 24/7 buoy network to guide ships from colliding with the world's last 400 North Atlantic right whales.

This work comes from a proposal of the Cornell University's Bioacoustic Research Program of finding new ways of improving the detection of these mammals through the audio signal of the buoys network. The proposal was made through a Kaggle competition named The Marinexplore and Cornell University Whale Detection Challenge [1] "Copyright © 2011 by Cornell University and the Cornell Research Foundation, Inc. All Rights Reserved".

Domain Background

Impressed by the working principle of Convolutional Neural Networks, I decided looking for uses beyond pure image classification. I also had been wondering if anything related to animals and whales could be done. I started looking in the internet and found several Kaggle competitions: one whale detection though images (Right Whale Recognition), and other to recognize the North Atlantic Right Whale call (The Marinexplore and Cornell University Whale Detection Challenge). Searching for applications of Convolutional Networks in sound recognition I found an entry related to the The Marinexplore and Cornell University Whale Detection Challenge. In [5] Daniel Nouri proposed to use ConvNets not just to go across the spectrogram of the Whale Calls, but try to recognize a pattern by simply looking at its image, like a human could. With this proposal he got pretty good results with a very straight forward approach. I decided to give it a try and look for most used ConvNets schemes and see their performance in this competition.

Problem Statement

Right whales make a half-dozen types of sounds, but the most characteristic one is the up-call.

This type of "contact call", is a little like small talk–the sound of a right whale going about its day and letting others know it is nearby. In figure 1 it is represented the spectrogram of an up-call which sounds like a deep, rising "whoop" that lasts about a second sound in [2], other calls in [3]). The goal of this

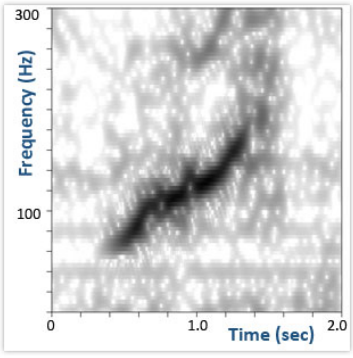

Figure 1: Spectrogram of a rigth whale up-call [2]

work it to be able to detect right whale up-calls, which is the most characteristic call of right whales, from the audio detected by the buoys deployed in the sea.

Datasets and Inputs

The dataset used comes from the competition and consists of 30,000 training samples and 54,503 testing samples. Each candidate is a 2-second .aiff sound clip with a sample rate of 2 kHz. The file "train.csv" gives the labels for the train set. Candidates that contain a right whale call have label=1, otherwise label=0. These clips contain any mixture of right whale calls, non-biological noise, or other whale calls [2, 3].

The training dataset is imbalanced, consisting of approximately 7000 Right Whales samples and 23000 of none Right-Whales samples. We could just balance the samples used or try to use an algorithm that penalizes this imbalance.

The audio is recorded after the buoys auto-detect the characteristic up-call, biasing the dataset to that kind of calls. Thus, it makes sense to detect only that kind of call, which is the most characteristic and most frequently emitted (details on the deployment of the buoys and how the recordings are made in [4]). This call (see figure 1) has a bandwidth of about 250Hz, fact that will help to reduce information processed to that range of frequencies.

Solution Statement

In order to perform the prediction I will try to implement well-known and widely-implemented models of neural networks.

One can be the LeNet-5, which is simpler structure than the one by Krizhevsky [7]. Figure 2 shows the structure of the network, composed of 2 convolutional layers, 2 fully connected layers and 2 subsampling or pooling layers.

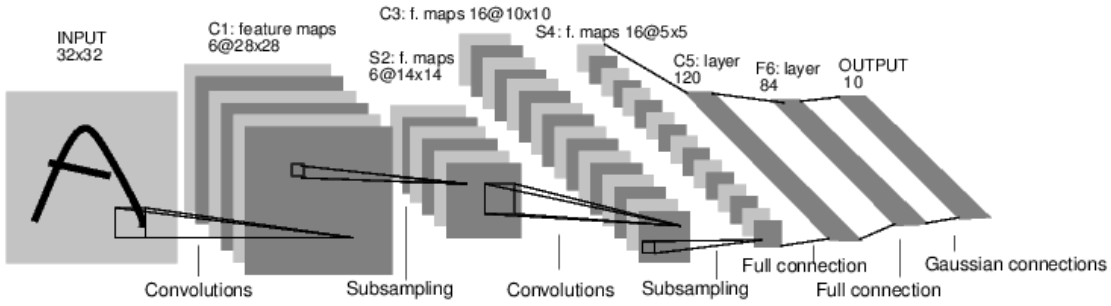

Figure 2: LeNet-5 structure [6]

Other structure I can use is the one by Krizhevsky [7]. Figure 3 shows the structure of the network, composed of 5 convolutional layers, 3 fully connected layers and 3 subsampling or pooling layers.

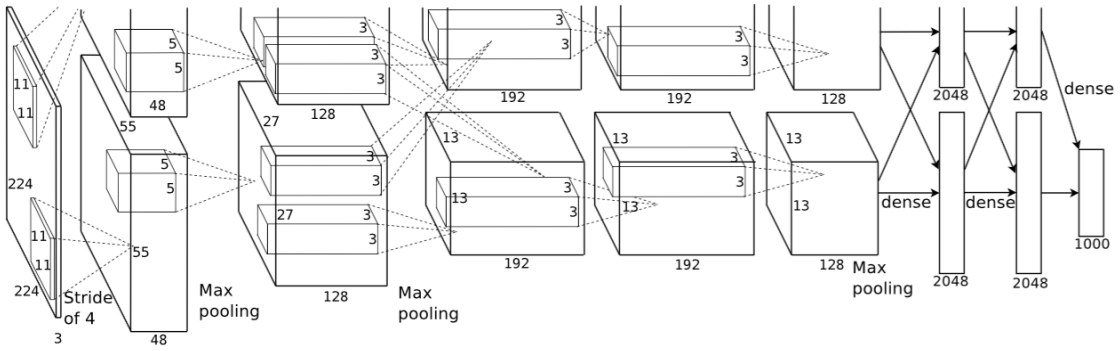

Figure 3: Network structure by Krizhevsky [7]

Benchmark Model

I will try to compare the performance of the most popular ConvNets (i.e. LeNet-5 proposed by Lecun [6] or the winner of the 2010 and 2012 ImageNet Large Scale Visual Recognition Competition (ILSVRC) proposed by Krizhevsky [7]) with the performance of the winning model of the competition which is based on Gradient Boosting (SluiceBox: Github) and the Daniel Nouri's model based on Krizhevsky's 2012 ILSVRC ConvNet model [7] (source), which first inspired this work.

The Area Under the Curve (AUC) (see the Evaluation metrics section) of these models in the public leaderboard was:

- SluiceBox: 0.98410 (1st position)
- Nouri: 0.98061 (6th position with 1/4 times the submission of the winner)

Nevertheless, I will not be able compare the performance of my models to these results. The reason is that I do not have the test labels and also, the public leaderboard data test used is slightly different for each participant. I will try two different approaches:

1. assuming that there are enough complete data samples (train dataset), trying to increase the accuracy as much as possible (this will be the main option)
2. assuming the predictions generated by the winning model as test labels and them as reference to compare our model with

Evaluation Metrics

The main evaluation metric for this project will be that used in the Kaggle competition, this is the **Area Under the Curve (AUC)**, where the Curve is the ROC curve.

The **receiver operating characteristic (ROC)** curve is a graphical plot that illustrates the performance of a binary classifier system as its discrimination threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.

The true-positive rate is also known as sensitivity, recall or probability of detection. The false-positive rate is also known as the fall-out or probability of false alarm.

The ROC curve is thus, the sensitivity as a function of fall-out. In general, if the probability distributions for both detection and false alarm are known, the ROC curve can be generated by plotting the cumulative distribution function (area under the probability distribution from $-\infty$ to the discrimination threshold) of the detection probability in the y-axis versus the cumulative distribution function of the false-alarm probability in x-axis (see figure 4)[8]

Another important measure can be the **error rate** vs iterations for a different batch sizes. The error rate used can be the percentage of wrong classified samples.

It can be also interesting to use the **confusion matrix**, which is a more detailed version of the ROC curve. The confusion matrix is a table that shows the predicted labels for each of the true input labels.

Project Design

1. sound samples exploration
2. spectrogram generation and image processing (contrast, appropriate dimensions...)
3. separation of dataset into training, cross-validation and test dataset and save into pickle
4. select ConvNets model and adjust parameters
    - define structure of the ConvNet adequate for my images: depth of layers, and stride and patch size of filters and pooling layers
    - AUC vs epochs (or training iterations), Error vs epochs, for different batch sizes
    - tune the model using regularization and decaying learning rate
5. compare the performance of winning model using the reduced version train and test dataset extracted form the train dataset

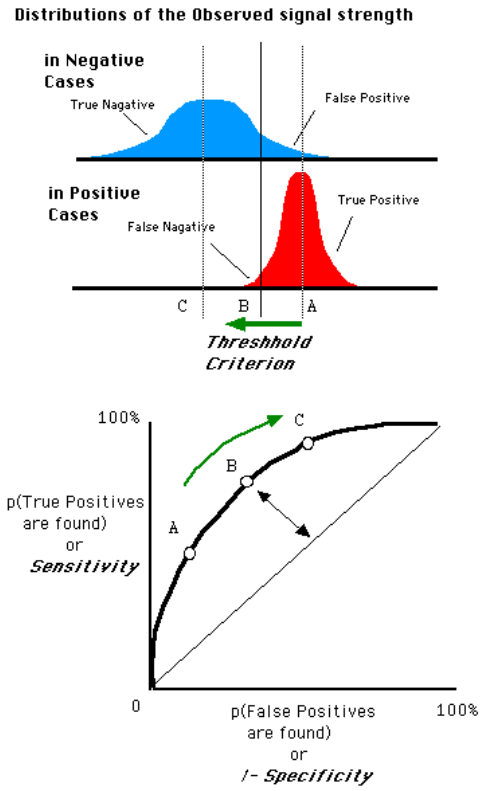

Figure 4: ROC curve graphic explanation [9]

References

[1] Kaggle. The Marinexplore and Cornell University Whale Detection Challenge. URL https://www.kaggle.com/c/whale-detection-challenge.

[2] Cornell Bioacoustics Research Program. Right Whale's Up-Call, Cornell Bioacoustics Resear, . URL http://www.listenforwhales.org/page.aspx?pid=432.

[3] Cornell Bioacoustics Research Program. More Right Whale calls, . URL http://www.listenforwhales.org/page.aspx?pid=442.

[4] M A McDonald and S E Moore. Calls recorded from North Pacific right whales (Eubalaena japonica) in the eastern Bering Sea. *Journal of Cetacean Research and Management*, 4(3):261–266, 2002. ISSN 1561-0713. URL http://www.afsc.noaa.gov/nmml/PDF/rightcalls.pdf.

[5] Daniel Nouri. Using deep learning to listen for whales — Daniel Nouri's Blog. URL http://danielnouri.org/notes/2014/01/10/using-deep-learning-to-listen-for-whales/.

[6] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998. ISSN 00189219. doi: 10.1109/5.726791.

[7] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet Classification with Deep Convolutional Neural Networks. *Advances In Neural Information Processing Systems*, pages 1–9, 2012. ISSN 10495258. doi: http://dx.doi.org/10.1016/j.protcy.2014.09.007.

[8] Wikipedia; the free encyclopedia. Receiver Operating Characteristic (ROC). URL https://en.wikipedia.org/wiki/Receiver_operating_characteristic.

[9] Wikiwand. Receiver operating characteristic. URL http://www.wikiwand.com/it/Receiver_operating_characteristic.