An Algorithm to Identify Blue Whale A calls from Underwater Recordings

Soumadeep Saha, Aditya Shankar Pal
Team - UG&Sons

Introduction

This rather interesting problem asks us to identify the presence (or absence) of "A-calls" of blue whales from under-water hydrophone recordings. With 25,946 labeled instances and a roughly even split between the classes, this problem is well suited for a deep learning based approach.

Our investigation showed that across the literature [1] and in similar past contests [2] the best approach has been to train an established image classification CNN model on a spectrogram representation of the audio data, and such methods are readily available [3][4]. Since this avenue is already well studied, we wish to try a different approach.

In principle all the information about whale calls is present in the raw signal, so it should be possible to train a deep neural network to perform well on this task with the time series data provided. To test this we designed our own network and trained it on the time series data. Our network has a micro F1 of 0.99 on a held out validation set and 0.985 on the test set, this landed us in the first spot amongst all duo teams, and an overall third position. In the coming sections we will briefly outline our technique involving model selection, data pre-processing and training before ending with a few concluding remarks.

Model Selection

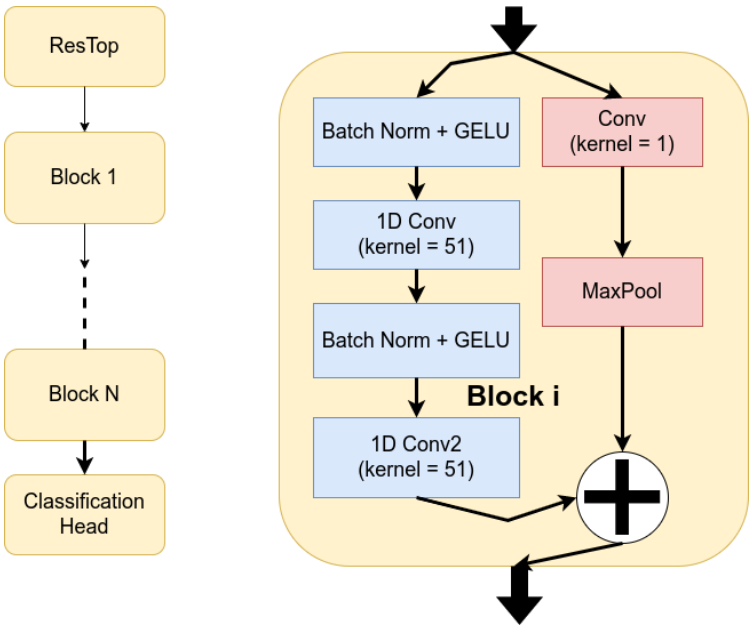

Fig. 1: Our 1D convolution based deep neural network with shortcut connections.

Although deep neural networks are universal function approximators, the inductive biases of a deep-learning model should be in tune with the properties of the data in question to aid in model

convergence and reducing training time. Since the data in question does not exhibit long term correlations (whale call at time T_0 does not influence call at time T_1) sequence models (LSTM, transformers, etc) are ill-suited. The data however does exhibit translational equivariance, thus pointing us to CNNs. In particular we used a 1D deep convolutional neural network for our detector.

However, since deep neural networks get harder to train with increasing depth, we employed shortcut connections [5] in our model to alleviate this problem. The number of channels is doubled every 4 blocks (starting from 1) and we downsample with strides = 2, every 2 blocks. In total we employ 20 convolution blocks (including Top). Our classification consists of two fully connected layers and the model predicts a single value between (0,1) with its sigmoid output. These hyperparameters were chosen via grid search.

We train our model with a weighted version of binary cross entropy loss, where every positive class instance is weighted twice as much as every negative instance [6]. This leads to models with slightly higher sensitivity, which is desirable since the target metric is Micro F1 which penalizes incorrect predictions of positive labels more severely.

Pre-processing

Since we used the time series data directly, our preprocessing pipeline is straightforward. We first resampled the data to 512Hz (since the frequencies of interest lie in the 70-90Hz band, this is well over the Nyquist limit to study relevant features).. This was done to cut down on computational time and memory. Following this we used a high-pass filter with cutoff at 65Hz and a low-pass filter with cutoff of 95Hz.

We ensured that each signal is exactly 32 seconds in length by either cropping the signal if it is bigger than 32s or filling up the shortfall with copies of the data to reach 32 seconds. Since the data is class balanced this won't lead to over/under sampling, and labels aren't affected by this procedure. Finally, we scaled the data by its standard deviation so that the gain is uniform across samples.

Training and Inference

We trained our model for 50 epochs with the AdamW optimizer and a scheduler that reduces the learning rate whenever a plateau is encountered. We saved the best model based on a held out validation set with 2000 instances.

We chose the positive prediction threshold by scanning over possible prediction thresholds, and choosing the one that led to maximum Micro F1 score.

Conclusions

Our deep neural network, designed from scratch, performs close to optimal (MicroF1 = 0.985 on the test set) with a novel approach that uses a pure time series signal. This stands as a testament to the power of deep learning where fantastical results can be obtained without the need for expert driven feature selection or complex preprocessing routines.

References

[1] An open access dataset for developing automated detectors of Antarctic baleen whale sounds and performance evaluation of two commonly used detectors - Miller, B.S. , et al. *Nature* 806 (2021). January 2021.
[2] Kaggle whale detection challenge - *https://www.kaggle.com/c/whale-detection-challenge*
[3] Submissions to kaggle whale detection challenge by TarinZ - *https://github.com/TarinZ/whale-detector*
[4] Kaggle whale detection challenge solution by jaimeps - *https://github.com/jaimeps/whale-sound-classification*
[5] Towards Understanding the Importance of Shortcut Connections in Residual Networks - Liu, T, et al - *https://arxiv.org/abs/1909.04653*
[6] Focal Loss for Dense Object Detection - Lin, T, et al - *https://arxiv.org/pdf/1708.02002.pdf*