CS583A: Course Project

Aditya A Kulkarni

December 1, 2019

1 Summary

I have participated in an inactive competition of identify individual whales based on images of their tails. The final model I chose is a deep convolutional neural network which takes 100x100 images as inputs and outputs 5 labels with highest probabilities.

I Implemented CNN using Keras and ran the code on Acer Predator Helios with Intel i7 CPU, 16GB memory and 1 NVIDIA GeForce GTX 1060 GPU with 6GB memory. The Performance is evaluated according to the Mean Average Precision @5.

My score in public leaderboard is 0.345 and in private leaderboard is 0.32. Since it is an inactive competition I was not able to submit my result for a leaderboard rank.

2 Problem Description

Problem. The problem is to identify whales based on images of their tails. This is a multi class classification problem.

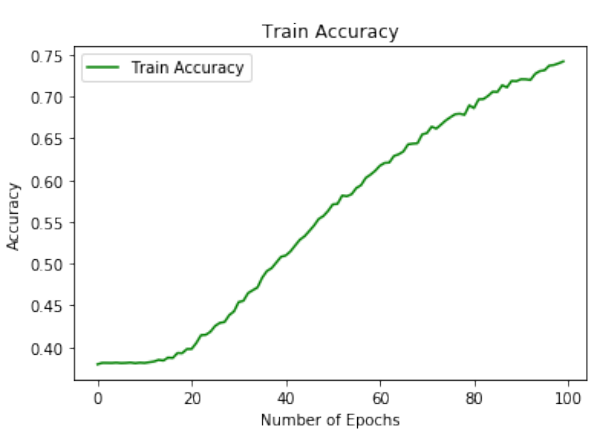
The competition is at `https://www.kaggle.com/c/humpback-whale-identification/overview/description`.

Data. The data is zip files of images. The number of training samples is 25361. The number of classes are 5005. The training set is not balanced as there are very few images for each class and a lot of images belong to one specific class: new_whale. There is a CSV file which maps each image in training set to a class. There is a test set with 7906 images.

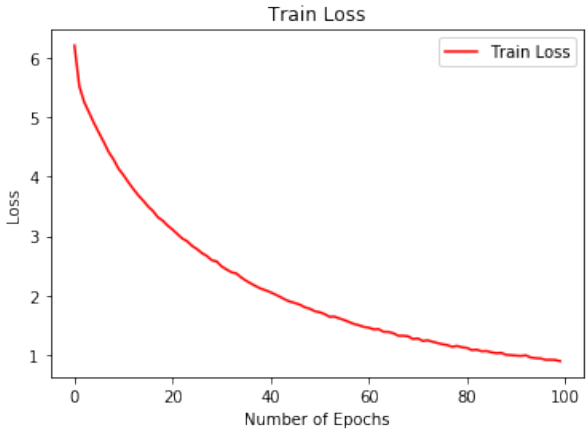
Challenges. The training data is inadequate and imbalanced. The number of images for each class are very few and a lot of images belong to a class new_whale which makes it imbalanced.

3 Solution

Model. The model I chose is 6-layer a standard deep convolution neural network, with batch normalization and dropout.



(a) The classification accuracy on the training set and validation set.



(b) The loss on the training set and validation set.

Figure 1: The convergence curves.

Implementation. I implemented a Simple CNN model using Keras with TensorFlow as the backend. My code is available at `https://github.com/BeardedAmbivert/Deep-Learning/tree/master/Project`. I ran the code on Acer Predator Helios with Intel i7 CPU, 16GB memory and 1 NVIDIA GeForce GTX 1060 GPU with 6GB memory. It took approximately 4 hours to train the model.

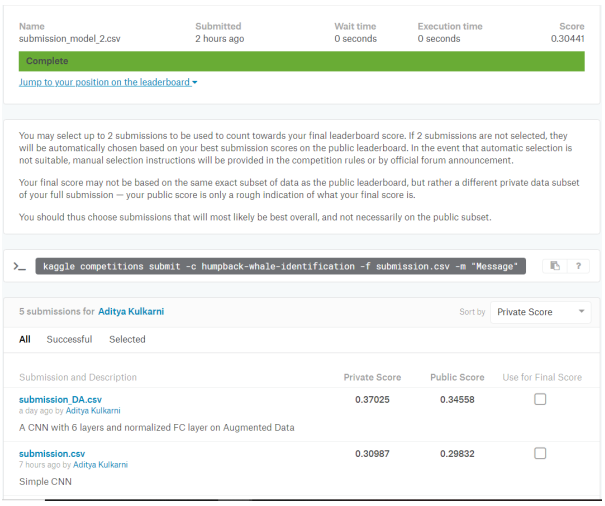
Settings. The loss function is categorical cross-entropy. The opitmizer is Adam. The number of epochs are 100. The batch size varied for each model, as the computation requirement increased. The batch sizes used are 1000 and 16.

4 Compared Methods

Fully-connected neural network. I implemented a 2-layer and a 6-layer convolution neural network. The width of the 2-layer network (from bottom to top) is 32, 64 and the width of the 6-layer network (from bottom to top) is a couple of layers with 16, 32 and 64. I applied dropout to the last layer and batch normalization to the first convolution layer for the 2-layer network. In the 6-layer network batch normalization is applied to each convolution layer followed by max pooling after every 2 convolution layers. The training accuracies for each network are 98.0% and 74.0% respectively.

Advanced tricks. Finally adopted model is a 6-layer CNN. We applied the following tricks.

- Data augmentation. Without data augmentation, the training accuracy is 98.0%. The model was over-fitting. With data augmentation, the training accuracy is 74.0%. Better generalization.



(a) Submissions

Figure 2: My submission score.

5 Outcome

I participated in an inactive competition. Using Data augmentation did not help much to improve my score.

My score is 0.37025 private leaderboard and 0.34558 in the public leaderboard.

The screenshots are in Figure 2.