Machine Learning Engineer Nanodegree

Capstone Proposal

by Tahsin Mayeesha

Proposal

Domain Background

Almost 50% of the world depends on seafood for their main source of protein. And most of the worlds high grade fish supply comes from Western and Pacific Region, which accounts for around $7 billion market. However, illegal fishing remains a threat for the marine ecosystem in these regions as fishermen often engage in overfishing and catching of protected species for deep-sea tourism such as shark and turtles. According to Fortune report on current usage of artificial intelligence in fishing industry , fishing operators in the pacific region typically sends a physical observer to accompany fishermen about 10 times out of 200 times in a year, however, this is clearly insufficient as there's no one to monitor what is going on in the other trips.

To combat the problem of proper monitoring, The Nature Conservancy , a global nonprofit fighting environmental problems has decided to create a technological solution by installing electronic monitoring devices such as camera, sensors and GPS devices to record all activities on board to check if they are doing anything illegal. However, even if having access to hours of raw footage is useful, according to TNC, for a 10 hour long trip, reviewing the footage manually takes around 6 hours for reviewers. On top of hectic conditions on a fishing boat, poor weather conditions such as insufficient light, raindrops hitting the camera lenses and people obstructing the view of fishes, often by choice, makes this task even harder for a human reviewer.

To automate this process, TNC partnered with Kaggle to ask machine learning practitioners to build a system that automatically detects and classifies fishes from the video footage data with a $150,000 prize to offset the costs involved in training deep convolutional neural network. The Nature Conservancy Fishery Monitoring competition has been featured in publications such as Engadget ,Guardian and Fortune.

The aim of this project is to build a convolutional neural network that classifies different species of fishes while working reasonably well under constraints of computation with help of transfer learning technique.

Problem Statement

The fish dataset was labeled by TNC by identifying objects in the image such as tuna, opah, shark, turtle, boats without any fishes on deck and boats with other fishes and small baits.

The goal is to predict the likelihood that a fish is from a certain class from the provided classes, thus making it a multi-class classification problem in machine learning terms.

Eight target classes are provided in this dataset : Albacore tuna, Bigeye tuna, Yellowfin tuna, Mahi Mahi, Opah, Sharks, Other (meaning that there are fish present but not in the above categories), and No Fish (meaning that no fish is in the picture).

The goal is to train a CNN that would be able to classify fishes into these eight classes.

Datasets and Inputs :

To create the dataset, TNC compiled hours of boating footage and then sliced the video into around 5000 images which contains fish photos captured from various angles.The dataset was labeled by identifying objects in the image such as tuna, shark, turtle, boats without any fishes on deck and boats with other small bait fishes.

The dataset features 8 different classes of fish collected from the raw footage from a dozen different fishing boats under different lighting conditions and different activity, however it's real life data so any system for fish classification must be able to handle this sort of footage.Training set includes about 3777 labeled images and the testing set has 1000 images. Images are not guaranteed to be of fixed dimensions and the fish photos are taken from different angles. Images do not contain any border.

Each image has only one fish category, except that there are sometimes very small fish in the pictures that are used as bait. The Nature Conservancy also has kindly provided a visualization of labels, as the raw images can be triggering for many people.

Solution Statement

As deep learning techniques have been very effective in image classification over the years, in this project, transfer learning along with data augmentation will be used to train a convolutional neural network to classify images of fish to their respective classes. Transfer learning refers to the process of using the weights from pre-trained networks on large dataset. Fortunately many such networks such as RESNET, Inception-V3, VGG-16 pretrained on imagenet challenge is available for use publicly.

Benchmark Model

Random choice : We predict equal probability for a fish to belong to any class of the eight classes for the naive benchmark. This submission yields 2.41669 log-loss in the Kaggle leaderboard.

K-nearest neighbor classification : A K-Nearest neighbor model was trained on the color histogram of the images with Euclidean distance as distance metric. This yields 1.65074 log-loss in the submission leaderboard.

A well-designed convolutional neural network should be able to beat the random choice baseline model easily considering even the KNN model clearly surpasses the initial benchmark. However, due to computational costs, it may not be possible to run the transfer learning model with VGG-16 architecture for sufficient number of epochs so that it may be able to converge.

So the reasonable score for beating the KNN benchmark would be anything <1.65074 even if the difference is not large considering running the neural network longer would keep lowering the loss.

Evaluation Metrics

The metric used for this Kaggle competition is **multi-class logarithmic loss** (also known as categorical cross entropy)

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

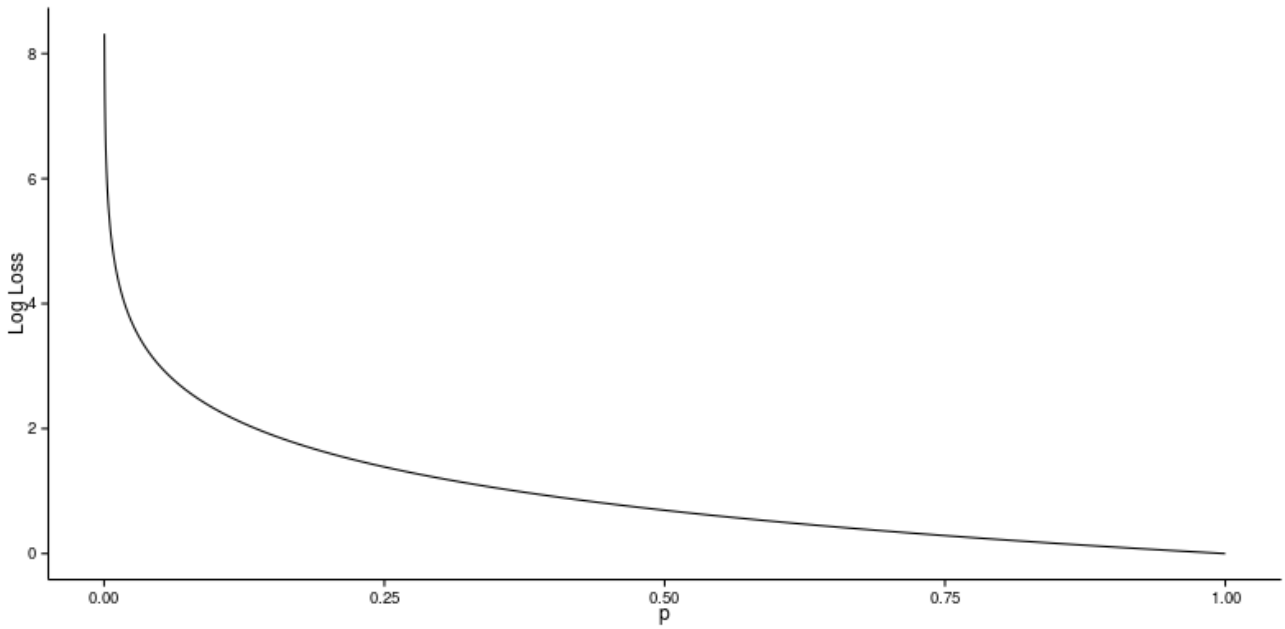
Here each image has been labeled with one true class and for each image a set of predicted probabilities should be submitted. N is the number of images in the test set, M is the number of image class labels, log is the natural logarithm, y_{ij} is 1 if observation i belongs to class j and 0 otherwise, and p_{ij} is the predicted probability that observation i belongs to class j.

The submitted probabilities for a given image are not required to sum to one because they are rescaled prior to being scored (each row is divided by the row sum). A perfect classifier will have the log-loss of 0.

Multiclass log-loss punishes the classifiers which are confident about an incorrect prediction. In the above equation, if the class label is 1(the instance is from that class) and the predicted probability is near to 1(classifier predictions are correct), then the loss is really low as $log(x) \to 0$ as $x \to 1$, so this instance contributes a small amount of loss to the total loss and if this occurs for every single instance(the classifiers is accurate) then the total loss will also approach 0.

On the other hand, if the class label is 1(the instance is from that class) and the predicted probability is close to 0(the classifier is confident in its mistake), as $log(0)$ is undefined it approaches $-\infty$ so theoretically the loss can approach infinity. In order to avoid the extremes of the log function, predicted probabilities are replaced with $max(min(p, 1-10^{15}), 10^{15})$.

Graphically[1] , assuming the i_{th} instance belongs to class j and y_{ij} = 1 , it's shown that when the predicted probability approaches 0, loss can be very large.



Project Design

- **Programming language** : Python 2.7+
- **Libraries :** Keras, Tensorflow, Scikit-learn, Opencv
- **Workflow :**
    - Establishing the baselines with random choice and K-nearest neighbors for comparison.
    - Training a small convolutional neural network from scratch for further comparison with transfer learning models.
    - Extracting features from the images with the pretrained network and running a small fully connected network 8 output neurons on the last layer to get predictions. Comparing it with running SVM on the extracted features.[2]
    - Fine tuning the pretrained network by choosing different optimizers and by training the network on this dataset from the convolutional layers instead of the dense layers as long as it's computationally inexpensive.
    - Optionally, comparing the performance of multiple pretrained networks. However, as finetuning them is computationally expensive, different pretrained networks can be compared at the feature extraction stage instead of direct comparison.

References :

---

1. http://www.exegetic.biz/blog/wp-content/uploads/2015/12/log-loss-curve.png "Log-Loss graph"↩

2. http://cs231n.github.io/transfer-learning/ "CS231, Andrej Karpathy's overview on Transfer Learning"↩