Progress Report 5

groveh

May 2019

Problem Description

Integration to turtle image data has been the primary goal of this project. With the network maturing in optimization as well as data augmentation, this report is dedicated to briefly discussing the results of the final network run on the turtle dataset. Approaches and solutions have been covered in previous reports, primarily report 4 for turtle dataset discussion.

Problem Approach

Data augmentation is a primary task for this dataset, as there are 4,518 usable (usable images were defined in the results section of report 4) images in total including testing and validation sets. MNIST consists on 70,000 total images and is a baseline standard dataset, therefore useful data augmentation is crucial to the success and reliability of running the network on the turtle dataset.

Solution

A combination of pre-processing and data augmentation allows for network training on turtles. Pre-processing loads images and annotations into memory and filters erroneous inputs, and data augmentation crops the image to the rotation corrected region of interest as defined by the annotation.

Challenges Faces

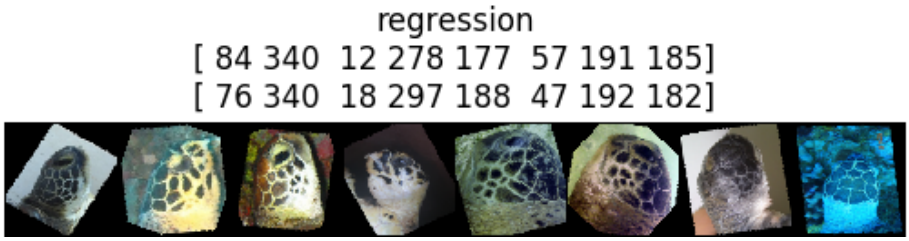
Additional challenges arose around normalization during testing. Means and standard deviations are computed during pre-processing, and used when loading each batch to normalize an image. Originally computed for each set (training, validation, and testing), an error arose where setting the model to either train or evaluate brought an inconsistency to these normalization values when applied to the images. Means and standard deviations cannot be computed over multiple sets of images due to laws of data snooping, however the error lies in the way PyTorch handles these modes. batch normalization uses tracked running statistics to use during batch normalization instead of those generated from the current batch data. This interfered with the resulting prediction angles, therefore inference mode has been temporarily until further investigation can be done

as to why a running mean and standard deviation lead to poor testing results. As a side note, this error has been isolated when, all else held constant, eval() was replaced with train() to result in significantly better results.

Results

The network after pre-processing and data augmentation performs at above satisfactory levels. Random examples of images and their predicted output were selected to display the process of rotating the images back to their inferred natural orientation. Below are various examples of a 360° regression algorithm being applied to the turtle dataset.

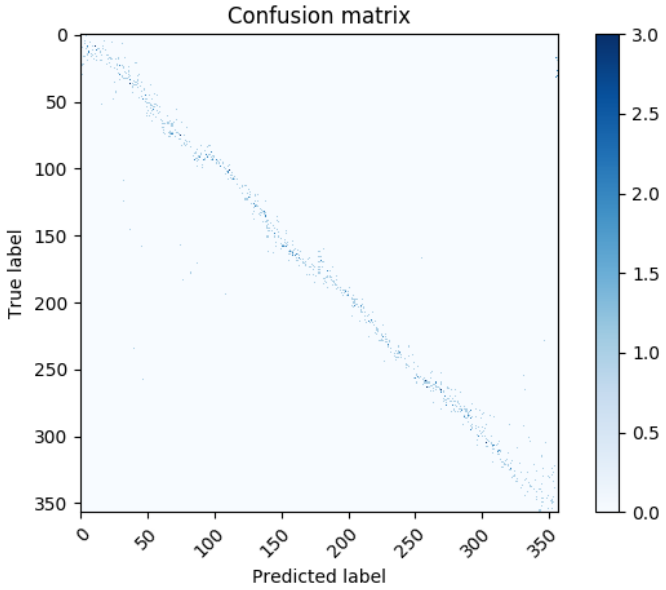
(a) Example 1 of predicted turtle images.



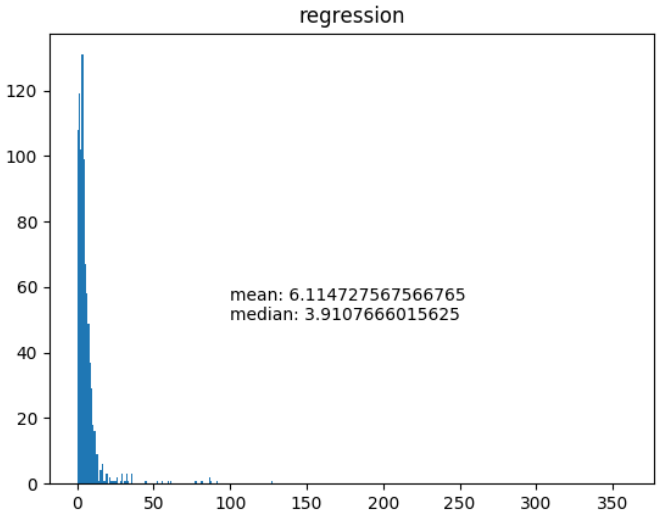
(b) Example 2 of predicted turtle images.

Figure 1: Two example output predictions of turtle head rotations. The top row of angles is the ground truth, and the bottom is the predicted rotation.

The above images show promising results. All predictions are within 20° of their ground truth, and the majority are within 10°. This is only a small subset of images, however they serve to visualize that small error is relatively acceptable. Below is the full error distribution in the form of a confusion matrix or true and predicted labels as well as a histogram of angle errors. The mean error is 6.11° and the median error is 3.91°, which means that there is a small tail of errors that extend to high degrees, but it is not significant.



(a) Confusion matrix of predictions to ground truth.



(b) Histogram of error distribution.

Figure 2: Two example output predictions of turtle head rotations. The top row of angles is the ground truth, and the bottom is the predicted rotation.

Distribution of error can be seen to congregate around zero with few instances that extend far beyond the mean, and the confusion matrix shows that the majority of predictions follow the ground truth diagonal.

Future Work

The issue presented about batch normalization during inference mode versus training mode of the model is a problem that needs to be addressed eventually. This could possibly involve modifying the way that the mean and stdev are applied during training or evaluation which is a process handled by PyTorch. Further investigation is needed to determine how to work with the difference between batch and running statistics.