Predicting Driver Behavior with Convolutional Neural Networks

Diveesh Singh
diveesh@*stanford.edu*

Problem

With the number of car accidents rapidly increasing, insurance companies need to ensure that they are keeping their prices and policies up to date. It can help to know the cause of accidents, so they can provide better coverage. By using Convolutional Neural Networks, we can train various models to look at pictures of drivers and predict what they are doing. The approach

The Dataset

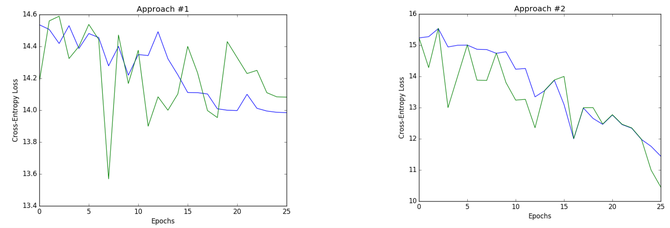
The data was obtained through State Farm Insurance via the Kaggle website. The dataset consists of about 4GB worth of photos, where each photo belongs to one of 10 classes. The classes are as follows: Safe Driving, Texting - Right, Talking on the Phone - Right, Texting - Left, Talking on the Phone – Left, Operating the Radio, Drinking, Reaching Behind, Hair and Makeup, and Talking to Passenger. The data is divided up into groups, where each group contains multiple frames of a timelapse

Initial Results

We performed training for 5 epochs on both models using the softmax approach and a reduced version of the training set. Using the accuracy metric provided below, we obtained a training loss of 14.387 and a validation loss of 14.399 with the first approach; the second approach gave us very similar results, which makes sense as 5 epochs of training is not a lot

Further Experimentation

We trained the both models from approach #1 and approach #2 for 25 epochs, with a training set of approximately 10,000 training examples, divided evenly from each class. We used a validation split of 0.2. After training for 25 epochs, approach #1 gave us a training loss of 13.985 and a validation loss of 14.082, which was not much better than the results obtained from 5 epochs. Approach #2 gave us a training loss of 11.445 after 20 epochs and a validation loss of 11.923. Below are the loss graphs



Model Experimentation and Formulas

Based on the dataset, there were 2 main approaches to designing a model. The first approach was a basic 10-layer CNN that was trained from scratch solely on the provided dataset, where each picture was treated as an individual training example. The second approach was to take a pretrained CNN (VGGNet-19) and then perform transfer learning on our dataset, still treating each picture as an individual training example assigned to a specific class.

Initially, both models used a softmax activation on the final layer

$$L_i = -log(\frac{e^{f_{y_i}}}{\sum_j e^{f_{y_j}}})$$

The accuracy metric that was used to evaluate the model was

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

The above loss function is very similar to the loss function for categorical cross entropy

Discussion and Future Work

Experimenting with the learning rate could prove fruitful, as training a model from scratch (approach #1) may require a decaying learning rate. Also, the dataset is not just a conglomerate of individual pictures, but is also divided up into segments, where each segment contains an individual frame of a video. Taking advantage of this by using techniques like Early Fusion, Late Fusion, and Slow Fusion[1] would also be fruitful

References

[1] Karpathy, Andrej. Large Scale Video Classification Using Convolutional Neural Networks