Machine Learning Engineer Nanodegree

Capstone Proposal

Mahmoud Nada

August 30, 2019

Sign language Letter Recognition

Domain Background

American Sign Language (ASL) is the primary language used by many deaf individuals in North America, and it is also used by hard-of-hearing and hearing individuals. The language is as rich as spoken languages and employs signs made with the hand, along with facial gestures and bodily postures.

A lot of recent progress has been made towards developing computer vision systems that translate sign language to spoken language. This technology often relies on complex neural network architectures that can detect subtle patterns in streaming video. However, as a first step, towards understanding how to build a translation system, we can reduce the size of the problem by translating individual letters, instead of sentences.

Problem Statement

Our problem is that deaf individuals in North America have always hard time describing what they want to someone who is not familiar with ASL so the goal is to build a model that can recognize ASL Letters and hopefully make it in real time so that it makes it easier for them to communicate in simpler and more efficient way,

Solving this problem with machine learning and computer vision is a great choice the field is getting bigger and smarter over the time so it will have plenty of room for modification and enhancement.

Datasets and Inputs

The data set is a collection of images of alphabets from the American Sign Language, separated in 29 folders which represent the various classes.

The training data set contains 87,000 images which are 200x200 pixels. There are 29 classes, of which 26 are for the letters A-Z and 3 classes for SPACE, DELETE and NOTHING, each 29 train classes has 3000 images. These

3 classes are very helpful in real time applications, and classification. The test data set contains a mere 29 images, to encourage the use of real-world test images.

This dataset is publicly available at Kaggle in this link:
https://www.kaggle.com/grassknoted/asl-alphabet

Solution Statement

In this project we use deep learning approach to tackle the problem of classifying the sign letter in the given image. Here, for this problem we train a Convolutional Neural Network and we will be also using transfer learning to see how the accuracy will be affected. Transfer learning is a process that focuses on storing knowledge gained while solving one problem and applying it to a different but related problem. There are some pretrained network on ImageNet dataset such as Inception-V3, RESNET 50, VGG-16, VGG-19 whose weights can be used for developing the solution of this problem.

Benchmark Model

For the Benchmark, we will use a CNN created from scratch using Conv2D layers with increasing numbers of filters to progressively detect more complex patterns. Between the Conv2D layers, we will insert MaxPooling and dropout layers to reduce the input dimensions and finally the fully connected Dense layer(s). We train this model on the training set and then evaluate the predictions made by the model on the test set using the loss function and accuracy score.

Evaluation Metrics

I will use categorical_crossentropy as loss function and accuracy score as evaluation metrics for this model and plot the training and testing accuracy through each epoch also I will make a plot for loss value of the model on each epoch to make good evaluation of the model.

Project Design

1. First, we import the required datasets for the sign classification.
2. Make some visualization like the distribution of various categories of sign letters over the training dataset, also visualizing samples of the dataset images.
3. Separate the dataset into training, testing and validation sets and do some required pre-processing like one-hot-encoding, normalization etc.
4. We create a benchmark model from scratch, using a mixture of Conv2D, MaxPooling2D, Dropout and Dense layers.

5. Train the benchmark model on training set and then evaluate its predictions using loss and accuracy scores.
6. Then, we create other model(s) with transfer learning approach using the pretrained models like Inception-V3, RESNET 50, VGG-16, VGG-19 (We create one or more models as computationally feasible). We extract the features from the images using pretrained network(s) and the use a fully connected dense layer.
7. We fine tune the model using different optimizers and adjusting the various parameters.
8. Finally, we compare our results with the result of benchmark model and decide whether our model made improvements or not.

References:

1. https://en.wikipedia.org/wiki/American_Sign_Language
2. https://www.kaggle.com/grassknoted/asl-alphabet
3. https://medium.com/@coviu/how-we-used-ai-to-translate-sign-language-in-real-time-782238ed6bf
4. Udacity's Deep Learning Lessons - MLND