Machine Learning Engineer Nanodegree

Capstone Proposal

Robert Cottrell
January 8, 2017

Proposal

Domain Background

Convolutional deep learning networks have had a dramatic impact on image recognition capabilities. Even simple models are able to make highly accurate predictions on datasets like the MNIST database of handwritten images.[1]

The MNIST images, however, are relatively simple. Recognition on images in real world conditions is more difficult. Even more challenging, however, is recognizing multiple interdependent objects.

Traditional approaches to recognizing multiple objects in an image require a multistage pipeline.[2] First the image is scanned to localize areas of interest. Then the relevant portion of the image is segmented for separate analysis. Finally, individual objects can be recognized from each segment and then assemble back together to identify the whole.

Problem Statement

The goal of this project, however, is to train a deep learning model that it able to recognize an entire sequence of digits in a number all at once.The model should be able to recognize arbitrary digits with a high degree of accuracy. Furthermore, the model should be able to be embedded within a mobile application for real time recognition from a live camera stream.

Datasets and Inputs

This project will use images from the publicly available Street View House Numbers (SVHN) dataset.[3] The contains images of house numbers from Google Street View images. The images are all of varying sizes and quality. The dataset is divided into three different groups: 33,402 training images, 13,068 test images, and 202,353 extra but less difficult images that can be used for additional training.

---

[1] LeCun, Yann et al, “The MNIST Database of Handwritten Digits.” Retrieved January 08, 2017, from http://yann.lecun.com/exdb/mnist/.

[2] Yang, Xuan et al. “MDig: Multi-digit Recognition using Convolution Neural Networks on Mobile.” Retrieved January 08, 2017 from http://web.stanford.edu/class/cs231m/projects/final-report-yang-pu.pdf.

[3] “The Street View House Numbers (SVHN) Dataset.” Retrieved January 08, 2017, from http://ufldl.stanford.edu/housenumbers/.

The SVHN dataset also includes metadata containing the ground truth label of the house number, as well as bounding box coordinates for each individual digit in the number.

Solution Statement

This project will attempt to recreate the results of Google's image recognition project as reported by Goodfellow et al.[4]

A convolutional deep learning network will be built that can be trained on the training and extra images from the SVHN dataset. The model will be used to make real time predictions from a live camera stream in a mobile application.

The deep learning network will be built using TensorFlow and the resulting model will then be used to power an Android application.

Benchmark Model

Goodfellow et al reported a transcription accuracy rate of 96.03% for their best model. This will be the aspirational target. An independent implementation was able to train a model to 95.4% accuracy.[5] This represents an ambitious but more attainable goal. Model accuracy will be judged based on predictions made on the SVHN test dataset images.

Evaluation Metrics

The evaluation metric for the model will be transcription accuracy on the test images in the SVHN dataset. Accuracy will be defined as correctly predicting full house number in the image. For the purposes of this project, there will be no partial credit given for identifying part of the number. The model must correctly predict not only the number of digits present but also correctly identify each of those digits.

Evaluation of the Android application will be more subjective. The app should be able to correctly recognize house numbers that are properly focused and centered in the camera frame. The app should also be able to identify when there are no digits in view.

Project Design

The first stage of the project will be to download and preprocess the SVHN images. The metadata for each image needs to be consulted to find the bounding boxes for each digit in the number. A new bounding box will then be calculated which will fully encompass the whole number. The images will then be cropped and resized to uniform size suitable for use by the model.

---

[4]Goodfellow, Ian et al. "Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks" *arXiv:1312.6082 [cs.CV] 2014*.

[5] "Multi-Digit Number Extraction from Images Using Deep Learning for Torch." Retrieved January 08, 2017, from http://itaicaspi.github.io/SVHN-Multi-Digit-torch/.

Identifying an unbounded sequence of digits will be computationally intractable. Instead the project will arbitrarily place a limit of five digits. While not sufficient, for example, to recognize telephone numbers, this limit is sufficiently high to identify most house numbers.

The network will consist of several convolution layers which may be followed by pooling or normalization layers, followed by fully connected layers, and finally terminate in six softmax classifiers. The first classifier will be trained to count the number of digits in the number. It should also be able to recognize when no digits are present or when the number consists of more than five digits. The remaining classifiers will be trained to recognize the individual digits of the house number in sequence.

A typical solution for a multiple classifier model would be to train the additional classifiers to detect the presence or absence of their respective classes. In this case it would mean that the digit classifiers would either detect the absence of a digit or recognize one of the 10 numerical values. Unfortunately, this presupposes independent classes, which is not the case here. This naive approach could yield inconsistencies between the counter recognizer and the digits recognizers.

Instead, the counter classifier will be wholly responsible for deterring the number of digits in the number. The individual digit classifiers will only be consulted if the counter has determined that a digit should exist. More critically, the unused classifiers should not contribute to back propagation while training the model. This is nonstandard behavior for TensorFlow and will require taking more control over the training process.

The final evaluation of the model will be determined by computing the transcription accuracy of the predictions made agains the SVHN test set. A separate validation set will be used to evaluate changes to the model's architecture and hyperparameters. The validation set will be chosen from the training set by using a shuffled and stratified split. This will ensure that the validation set will more closely match the "harder" quality of the test set. The images from the extra set will be used for training but will not be used for validation.

Once the model has been fully trained and evaluated, the weights will be frozen and extracted to a form that can be used for inference. TensorFlows demo Android app will be adapted to use the trained model to predict numbers from a live camera stream.