Multiple Digit Recognition Capstone Proposal.

Machine Learning Capstone - Michael Amundson - April 15, 2017

Domain Background

Optical character recognition is an important technology for converting images into characters that are recognized by a computer. This technology has a wide range of applications, from recognizing the characters in images of scanned documents to the new google translate app that can use a phone's camera to read written text and then translate it into a different language. Character recognition has been attempted with machine learning techniques such as linear classifiers, SVM, KNN, and neural networks (LeCun 1998).

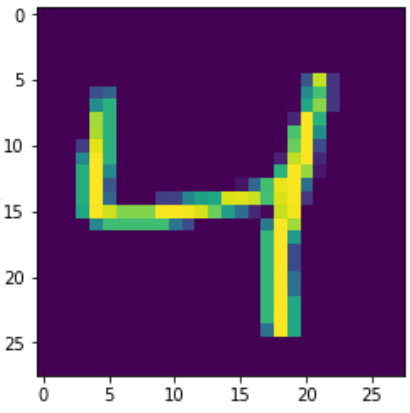
Some of the best results for character recognition have come with convolutional neural networks such as the LeNet architecture by Yann LeCun trained on MNIST data and Goodfellow's deep CNN trained on Google street view house number (SVHN) data (Goodfellow 2013).

Problem Statement

I want to build a classifier that can recognize the digits in a given image. This program will need to determine the location of up to five digits in an image and classify each digit(0-9). This classifier will also need to be computationally cheap enough to train on my pc in a reasonable amount of time and able to make predictions quickly.

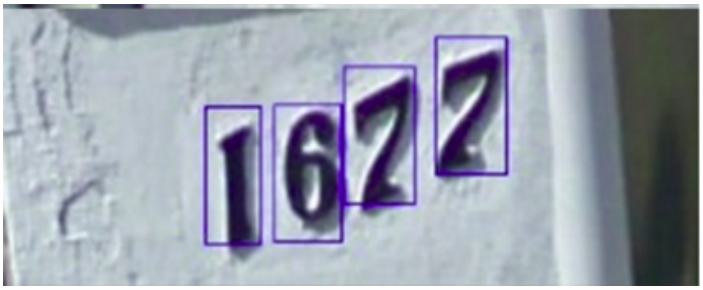
Datasets and Inputs

I will use two datasets to train networks and see which gives the most satisfactory results. First I will train on the MNIST data. Since MNIST is single characters I will combine 1-5 characters to form a sequence filling the remaining space with null. MNIST is also single channel data so I will not have to deal with the complexity of a color image or image resizing when training.



Sample MNIST Character

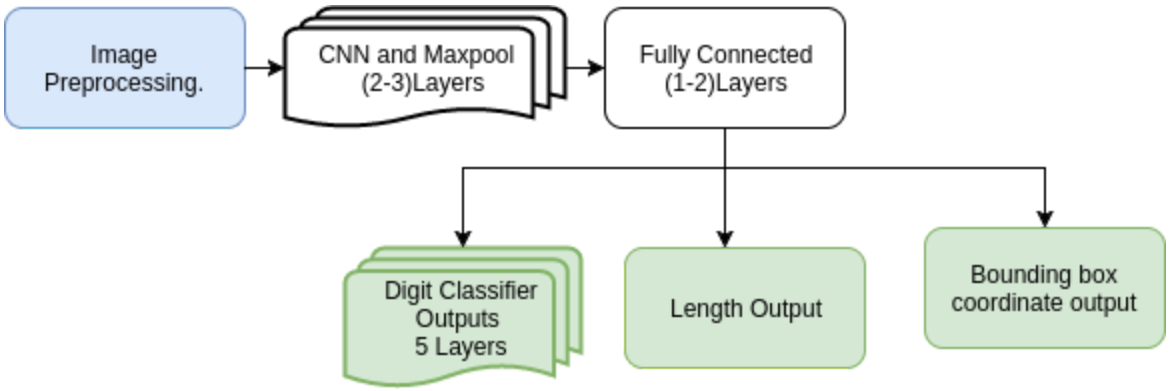
To develop the ability to classify color images that are not lined up in a row and are not all the same size I will also train on the Google SVHN dataset (Netzer 2011). This dataset has three channels and is made up of photos taken of house numbers with bounding boxes that give the location of the digits.



Sample SVHN Image

Solution Statement

My proposed solution is to build a convolutional neural net that can take the input images and separately classify each digit and provide the location of each digit. Each possible digit in the sequence will have its own output layer with softmax activation. There will also be an output layer for the length of the sequence. Finally the localization problem will be solved by an output layer that predicts the bounding box coordinates (top, bottom, left, right) of each digit. The model will minimize combined loss from each of the layers.



Benchmark Model

The Goodfellow model will be the benchmark. Their model was a CNN that recognized the digits in SVHN data. Their model consisted of eight convolutional layers, one locally connected layer and two fully connected layers. They used six softmax output layers, one for each digit and one for the length of the sequence. They did not feed the location of the images to the network so the network had to come up with its own spatial representation. The model by Goodfellow et al achieved 97.84% character accuracy and a 96.03% sequence accuracy on the SVHN test set.

Evaluation Metrics

I will evaluate my model's classification abilities based on the sequence accuracy defined as the number of sequences where every digit was predicted correctly divided by the total number of sequences. I will try to come as close to Goodfellow's 96.03% as I can however my goal is also to make a network that has less parameters and only 3-5 hidden layers compared to Goodfellow's 11. Since I am focusing on building a model with lower computational cost I do not expect to achieve their level of accuracy on the SVHN data.

To evaluate the localization abilities of my model I will use average Intersection Over Union (IOU). This metric divides the intersection of the predicted bounding box and the true bounding box by the union of the predicted bounding box and the true bounding box. I was not able to find a published benchmark for this dataset but I did find the heuristic that IOU's greater than 0.5 were generally considered good (Rosebrock 2016). I will try to achieve average IOU's for all of the non-null digits that are above 0.5.

Project Design

Languages and Libraries

Python2 - Programming Language
Sklearn - Machine learning library
Keras - Library to make it easier to build neural nets with tensorflow.
Tensorflow - Machine learning library for executing operations with neural nets.

Preprocessing

The MNIST data are all 28x28 characters and the characters take up a significant portion of each image so the only preprocessing needed will be to normalize from 0-255 to 0-1 and concatenate from 1-5 characters together.

The SVHN pictures are multiple sizes so they will need to be cropped and resized. I will crop the image to the smallest rectangle that contains all the individual digit bounding boxes. I will then resize the image to 54 x 54 pixels for input into the CNN. Since I am training on images that have already been cropped localization will be limited to within the cropped area and not the original SVHN image.

Proposed CNN architecture.

Convolutional Layer, 16 filters, 5x5 Window, 1x1 stride, relu activation.
Maxpool(2x2 pooling, 2x2 stride)
Convolutional Layer, 32 filters, 5x5 Window, 1x1 stride, relu activation.
Maxpool(2x2 pooling, 2x2 stride)
Convolutional Layer, 64 filters, 3x3 Window, 1x1 stride, relu activation.
Fully connected 256 node layer.
Fully connected 128 node layer.

Output Layers each connected to the last fully connected hidden layer:

Five fully connected 11 node layers with softmax activation (one for each digit we classify).
Fully connected 5 node layer with softmax activation to predict the length of the sequence.
Fully connected 20 node layer to predict top, bottom, left, and right of each digit's bounding box.

Loss will be calculated using the combination of categorical cross entropy for the output layers with softmax activation and mean squared error for the bounding box predictions.

References

Ian J. Goodfellow, I. Bulatov, Y. Ibarz, J. Arnoud, S. Shet, V. (2013). Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. Technical Report, arXiv:1312.6082

Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, Proceedings of the IEEE, 86(11):2278-2324, November 1998

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, Andrew Y. Ng Reading Digits in Natural Images with Unsupervised Feature Learning NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011.

Rosebrock, Adrian. "Intersection over Union (IoU) for Object Detection."PyImageSearch. N.p., 27 Sept. 2016. Web. 27 Apr. 2017.