Machine Learning
Multi Digit Recognition Project

Gaurav Narula *(2013B3A7516G)*, Devashish Deshpande *(2013A7PS122G)*, and Apurva Bhandari *(2013B1A7868G)*

Abstract—The aim of this project is to train a model that can decode sequences of digits from natural images. The model will be trained on the The Street View House Numbers (SVHN) Dataset, a public subset of street numbers captured by Google Street View's cars. The problem has been addressed by researchers at Google [2] and our approach is inspired by theirs.

Keywords—Digit-Recognition, SVHN, CNN, Keras

1 Introduction

Given an image of arbitrary size and an indication about the position of the street number in this image, to what extent is a model able to recognize it and identify correctly each of its digits? How accurate can it be? The objective of this project is to bring a decent solution for a subset of it, a street number, which is a sequence of up to 6 digits. Our objective is to develop a sound theoretical and practical understanding of Convolutional Neural Networks, and to gain the ability to implement training at scale (deep model with a lot of data) with Keras.

The final model was trained with a considerable amount of (augmented) data on an AWS g2.2xlarge instance.

April 22, 2017

2 Analysis

The SVHN dataset provided consised of pictures of various sizes and resolutions of street numbers in png format. A CSV file contained the metadata for all the images, denoting the bounding box coordinate for each digit in every file.

FileName	Digit	Left	Top	Width	Height
1.png	1	246	77	81	219
1.png	9	323	81	96	219
2.png	2	77	29	23	32
2.png	3	98	25	26	32
3.png	2	17	5	8	15
3.png	5	25	5	9	15

2.0.1 Metrics

A street number is a sequence of up to 6 digits, ranging from 1 to 999999. That would be too many classes for one classifier. We decided to use an approach suggested by Goodfellow et al.[2], and ended up with 6 outputs, each output classifying a digit between 0-9 or no digit, thereby resulting in 11 classes per output for a total of 66 classes.

The model was evaluated on its accuracy, which is the ratio of correct predictions over all predictions. A prediction is a street number value. To predict this value, the model must predict a value for each digits.

3 Algorithm and Techniques

As written above this project follows the approach similar to that described in Goodfellow et al, 2014 [2]. Taking the scarcity of computational resources into account, we decided to use the architecture described below:

1) Input
2) CONV1-48-5 → RELU → BATCHNORM → MAXPOOL-2
3) CONV2-64-5 → RELU → BATCHNORM → MAXPOOL-2 → DROPOUT
4) CONV3-128-5 → RELU → BATCHNORM → MAXPOOL-2 → DROPOUT
5) CONV4-160-5 → RELU → RELU → BATCHNORM
6) CONV5-192-5 → RELU → BATCHNORM → MAXPOOL-2 → DROPOUT
7) CONV6-384-5 → RELU → BATCHNORM → AVGPOOL-2
8) CONV7-768-5 → RELU → BATCHNORM → AVGPOOL-2
9) Flatten
10) 6 Dense layers of 11 depth each for outputs

Convolutional layers are motivated by the fact that an image is a composition of smaller but meaningful features. Since those filters convolve on the whole image, every pixels affects the weights of the filter (weight sharing). Activation functions, RELU in case of our model, introduce non linearities in the network. Without activations, neural nets would be combinations of linear or polynomial regressions, and adding non linearities make the features extracted more expressive. Pooling replaces the output of the convolutional layer with the max (Max pooling) or average (average pooling) of neighboring values.

3.1 Methodology

3.1.1 Metadata Preprocessing

The test data is bundled with a csv file containing the coordinates of top left and bottom right corner (bounding box) of each digit in an image. We decided to transform the data into a more readable form with DigitLabel modified to a list of digits in an image in the order they appear in the csv. The coordinates are combined to get an overall bounding box that spans over all the digits in the sequence.

For bounding box of each image using the bounding box given for each digit, we did the following:

1) For left,we took the minimum of the left value of all the digits in the image.
2) For top, we took the minimum again.
3) For right, max of left+width of each digit.
4) For bottom, max of top+height of each digit.

3.1.2 Data Transformation

We've preprocessed the data as follows:

1) Rotate the image by 4 degrees.
2) Increase the bounding box by 30% and crop around it.
3) We resized it to 37*37*3
4) We cropped out a random portion of maximum size
5) We then resized to 32*32*3. (We tried on 64*64*3 initially but looking at the images we realized that most of the bounding boxes had dimensions much smaller than 64*64. Hence we decided to try 32*32*3 for better accuracy.)

Digits of each image were then converted to One Hot Representation for calculating the loss later with the output of the CNN.

3.1.3 Validation

The validation set is a random extraction of 10% of the training data. In a later attempt, we removed the split and trained on all of the data.

4 Result

The architecture was trained with the following parameters

1) **Optimizer:** Adam with learning rate 0.0001
2) **Batch size:** 16
3) **Dropout:** 0.25 on layers with Pooling

Training was stopped after 7 epochs and an epoch took around 400s on an AWS g2.2xlarge instance.

4.1 Accuracy

With a 0.1 validation split and 64x64 images: 80.2%
Without 0.1 validation split and 32x32 images: 80.4%

5 Conclusion

We got a maximum accuracy of 80.4% with seven convolutional layers in our Neural Network. Perhaps better results could have been obtained by adapting a localization and detection approach where in a network would have figured out the bounding boxes for each digit and then the classifier would have determined the digit in each bounding box.

Acknowledgments

We would like to thank our professor Dr. Ashwin Srinivasan for providing us with the opportunity and the motivation to do a project of this kind. We'd also like to thank Rajat Agarwal for helping us throughout the project.

References

[1] The Street View House Numbers (SVHN) Dataset:
http://ufldl.stanford.edu/housenumbers/

[2] Ian J. Goodfellow, Yaroslav Bulatov, Julian Ibarz, Sacha Arnoud, Vinay Shet (2014). Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks

[3] CS231n: Convolutional Neural Networks for
Visual Recognition
http://cs231n.github.io/convolutional-networks/conv