Individual Final Report

Bill Grieser, Group 7

Introduction and Overview

The project was to work with the Street View House Numbers dataset for a classification task. We decided to each build and train a model in a different framework so we each got experience with building and training a convolutional neural network. I wanted to explore PyTorch's autograd function, and so I asked to do PyTorch and because it was a framework we had used in class extensively, try to find a deeper use for it. I decided to investigate removing the foreground digit from the image.

My Individual Work

My work is captured in the group report in the PyTorch section and the section on the digit removal task. The way we divided up the work allowed us to work in parallel and for each to have soup-to-nuts responsibility for one model, and the PyTorch work was mine.

I also created the preprocessing scripts to download and turn the SVHN data into pickles. This was made available to the others but I think everyone ended up needing their own preprocessing because of their framework.

Results

The results are given in the main report in the PyTorch and the Digit Removal from Images sections.

Summary and Conclusions

I learned that:

- PyTorch is very powerful. It was very easy to set up an optimization for the Fill network based on the RBF network.
- The CNN performed much better on the GPU than on CPUs machines. In fact, it would not have been possible to run in a timely manner without the GPU.
- Dropout is an important tool to use to combat overfitting. In the past I was worried about using high dropout factors like 0.5 but after this project and class, I understand better how to tell overfitting is happening and why dropout is not to be feared.
- I learned how to get to the internals of the PyTorch network to do interesting things like play with the gradients and inspect the feature maps.
- To calculate the inputs to the fully connected layer, I had to learn how the convolutional and pooling layers reduced their input and what output they produced. I automated the calculation of the input parameters in my CNN network once I understood this.

Improvements (if I had to do the project again):

- On the digit removal, I would use a different strategy to resize the images. The key pixels were determined using the same image size used by the prediction network, which then had to be translated to the original size, and I am fearful that the resize can cause pixels to get misidentified after the transform back to the original size. Perhaps I would use the basis vector transforms we learned at the beginning of the year if I could do it over.
- Smaller images are not performing well on the digit removal. I think the method I settled on to identify the key pixel (mean-std is in the most populated bucket) is promising, and I think the number of buckets to create should be tied to the image size -- the bigger the image, the more buckets.
- If I could I would like to create some training and test data and have the key pixels identified using a learned method. I did not have time to do this for this project but I initially considered it. I was confident that the gradients would lead me to the right answer -- confidence born of naivete, perhaps, and so I had to start over with the feature maps. There is still remnants in the code of attempts to use the gradients. I think the ultimate solution would be to use both gradients and feature maps and have a supervised network figure out how to make it work.

Internet Code

I used very little internet code directly. The main case was the code that read the metadata out of the matlab files that game with the data. (The matlab file version was too recent for SciPy to be able to read). A kind soul on Stack Overflow had posted code on how to read it for SVHN. I used 27 lines of this code. I also used about 20 lines from the PyTorch section of the Deep learning repoout of the 1753 lines of new code:

47/1753 = 2.6%

Also I reused and modified approximately 235 lines (not included in the 1753) code that I had written previously.

Bibliography

Key works that I relied on:

Alilou, V., & Yaghmaee, F. (2015). Application of GRNN neural network in non-texture image inpainting and restoration. *Pattern Recognition Letters*, 24-31.

Hagan, M. T., Demuth, H. B., Beale, M. H., & De Jesus, O. (n.d.). *Neural Network Design, 2nd Edition.*

Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., & Ng, A. Y. (2011). Reading Digits in Natural Images with Unsupervised Feature Learning. *NIPS Workshop on Deep Learning and Unsupervised Feature Learning.* Retrieved November 18, 2018, from http://ufldl.stanford.edu/housenumbers

Shperber, G. (2017, August 27). *Background removal with deep learning*. Retrieved December 1, 2018, from Towards Data Science: https://towardsdatascience.com/background-removal-with-deep-learning-c4f2104b3157