Deep Learning for Computer Vision: Final Project

Computer Science: COMS W 4995 006

Proposal: March 22, 2018

Presentations: April 19, 24, and 26, 2018

Report: April 26, 2018

Project Overview

The final project is one of the most important and, hopefully, exciting components of the course. You will have the opportunity to develop a deep learning system of your own choosing. You are free to select whatever framework (Tensorflow, Theano, Caffe) or wrapper (Keras, TFLearn) you like, but you need create a report on your project in a Jupyter notebook. You are also free build on publically available models and code, but your report must clearly give attribution for the work of others and must clearly delineate your contributions.

Project Proposal

The project description should include the title of the project, participants, a description of the objectives of the project, and a plan for how the project will be completed. The description of the objectives should include modest predictions of the success of the project. The plan for completion should include a description of the training data and how it will be obtained, a discussion of what deep learning framework will be used and why, and a rough description of the planned network architecture.

You are permitted to work together on a project in groups of two or three, but group size must not exceed three participants. For group projects there must be a clearly delineated division of labor: you should state in the project description and project report who was responsible for which portion of the project. Each student must hand in a separate report. (Students will not necessarily get the same grade for the same project.)

You should mention whether you are simply re-implementing what others have done before but applying to new data or whether you are attempting to do something new to the best of your knowledge. Creative and original projects will be judged more kindly than those that are rehashing something in the existing literature. And projects that include a component in which data is acquired/curated into training and validation sets will be veiwed more favorably than those that simply download an existing data set such as CIFAR-100.

As this is a computer vision course it is expected that your data will be visual, but excpetions might be made if the student is enthusiatic and persuasive enough. The most straightforward project would be to build a system that classifies images into catogories. A more difficult project might be to build a system that detects and localizes a type of object within an image. A still more complicated project might involve joining a ConvNet with an LSTM for a problem (like image captioning) that requires vision and language. But again, creative and originnal projects will be judged more kindly.

It is important to scope your project so that you get some working results. Project reports that say "I tried this and this but nothing seemed to work..." are discouraged. Above all, you should demonstrate end-to-end fluency in the basics of deep learning.

I cannot wait to see the results. Good luck!

Project Presentations

To allow students to present their work in three class periods, each student will have only 3 minutes, not a second more. We will be strict about the timing, so you should practice your presentation. The key here is to get across three things: what you did, how you did it, and how well it worked. Students working in groups of two will get 6 minutes and groups of three will get 9 minutes.

Project Reports

The report should be done as a Jupyter Notebook. The report should be a complete description of the objectives of the work, the methods used to solve the problem, experimental evidence of a working system, the code, and clear delineation of what you have done vs. what you are leveraging that others have done. If you have used the work of others YOU MUST INCLUDE ATTRIBUTION by citing this work inline and as part of a "bibliography" at the end. You should describe what worked, what did not, and why. If you are working in a group you need to submit your own report and this report should be clear about what your individual contribution was.

Title: Understanding Convolutional Neural Networks

Participants

This will be an individual project.

Objectives

I want to attempt to understand how CNNs work in the domain that is better understood from a traditional Computer Vision perspective. While this is too ambitious, I am going to focus on CNNs used for Visual Classification.

The "Computer Vision" Block

I propose to do the above by introducing a "Computer Vision Block" (CV-Block) within the Network. This CV-Block would be based on an a Computer Vision algorithm that is converted to an implementation that is backpropogable and hence, can be learnt. To explain this further, the CV-Block will have three components: (1) Transfer Image to feature space based on a deterministic Vision algorithm. (2) A layer which "selects" features from all the available space. (3) Transfer feature space back to the Image that can go through the rest of the network. This CV-Block can be introduced between any two layers of existing CNN networks used for Visual Classification. The weights in the "selection" layer of the CV-Block will help understand the features that were important for the said task.

The task mentioned above is still unclear in terms of how it would fit into implementation. I propose to start with a simpler analysis based on Fourier Transforms to understand the frequencies that are important for visual classification. This is the first basic step that I would like to start from. In the next section, how this is doable theoretically, and then I'll proceed to explain how I propose to do it using PyTorch.

Implementation Details

Dataset: I intend to start with CIFAR-10 dataset and move towards a bigger dataset once my architecture is better understood.

Architecture of the network with the Fourier-Block

Let's consider an example CNN Network of 6 CNN layers and 2 Dense layers including the input and output layers. The CV-Block, which in this case is the Fourier-Block, is placed between CNN layers 2 and 3. The layer wise operations would be as below:

1. Layer 1: Input CNN layer with ReLU activation
2. Layer 2: CNN layer with ReLU activation
3. Fourier Block: a. Fourier Transform with the resolution being the same b. 4 attention based unit to select four regions within the fourier transform c. Inverse Fourier Transform to an image of the same resolution
4. Layer 3: CNN layer with ReLU activation
5. Layer 4: CNN layer with ReLU activation
6. Layer 5: CNN layer with ReLU activation
7. Layer 6: CNN layer with ReLU activation
8. Layer 7: Fully Connected layer
9. Layer 8: Output layer, fully connected with number of units same as the num of classes for visual classification.

The layers 3a. and 3b. together could explain four of the most important regions in the frequency domain and the accuracy obtained with the network can explain the contribution of these frequencies to the maximum that the network is able to learn.

To evaluate, that the Fourier-Block can indeed figure out if the network worked well, I will be using the above mentioned architecture without the 3b. selection process. This should ideally be the same as the CNN without the Fourier-Block. This makes a base case argument for the credibility of the method.

Implemeting the Fourier-Block and feasibility analysis

My language of choice is PyTorch. It would allow be to write custom functions with the forward and backward propogation required for the project.

The Fourier Transform can be implemented using DCT Matrix, and hence is a linear transformation. Thus, it is also differentiable and hence, backpropogable. The same is true for the Inverse Fourier Transform. The complication in this case would be to figure out a way that the network can handle complex numbers because the Fourier Transform cannot be used without Phase information. It will also be important to analyse the effects of quantization. These can be analyzed based on my base case scenario of eliminating 3b. from the network.

The 3b. layer can be implemented based on the the networks in object detection that make use of masks for localization[1]. Another method to do that would be use attention based on networks like "Deep Recurrent Attentive Writer (DRAW) [2]"

Overall, I expect the implementation to be feasible even if it is too complicated. I am unsure of what the results will yield but I hope to gain some understanding of what the network learns in the Fourier Domain.

References

[1] He, Kaiming, et al. "Mask r-cnn." Computer Vision (ICCV), 2017 IEEE International Conference on. IEEE, 2017. https://arxiv.org/pdf/1703.06870.pdf (https://arxiv.org/pdf/1703.06870.pdf)

[2] Gregor, Karol, et al. "DRAW: A recurrent neural network for image generation." arXiv preprint arXiv:1502.04623 (2015). https://arxiv.org/pdf/1502.04623.pdf (https://arxiv.org/pdf/1502.04623.pdf)