Deep Compression

An End-to-End Deep Learning image compression pipeline using Convolutional Neural Networks

Qasim Wani, Rohit Selvam, Aman Mathur

I. Problem:

90% of all data in existence was created in just the past 2 years. And this pattern continues to be true as more and more people have access to the internet. However, data storage and infrastructure isn't scaling at the same rate as exponential data growth. To solve this problem, data compression seems to be a viable solution in representing large chunks of data into smaller representations. Techniques such as Variable Length encoding and Huffman Encoding all boils down to one fundamental principle in data compression: entropy - average level of uncertainty in a stochastic process. However, these techniques are often outdated in terms of computational complexity and availability to highly powerful algorithms and paradigms such as Deep Learning. As the principal focus of this class is around image processing, we've decided to tackle the problem of image compression using a specific Deep Neural Network - Convolutional Neural Network which preserves spatial relationship (translation invariance) between pixels by learning image features using small squares of input data. Another reason boils down to Kolmogorov complexity with respect to model size. Unlike a standard fully connected layer, CNN's have the ability of sharing parameters where a feature detector that's useful in one part of the network is probably useful (stochastically speaking) in another part of the image when utilizing the Convolution (cross-correlation) property. This makes the network to have fewer parameters, thereby reducing model size making it vital for tasks such as compression.

II. Approach

We've based our initial model on the image compression paper as referenced. We'll be developing the following: 2 Convolutional Neural Networks and 1 image codec representation for encoding and decoding the compact representation produced from the network. The pipeline works as follows: an image is passed as an input to the first CNN model, ComCNN which consists of 3 layers utilizing a combination of activation functions such as ReLu. This combination serves the purpose of patch extraction in the earlier layers and representation learning in the latter layers over overlapping patches from the input image. The output of this, say θ_1 is then encoded using an image codec such as JPEG, JPEG 2000, or BPG. This is then decoded using the same image codec and then the output of the decoded image is passed onto the reconstruction CNN network, RecCNN which consists of 20 layers consisting of convolutions, max pooling, ReLu, and batch normalizations. Upon passing through the network, an output θ_2 is generated which is then passed through an image interpolation technique such as linear or bicubic interpolation to unsample the image to original size. The underlying assumption is that Reconstruction network is monotonic w.r.t to the optimal input of RecCNN. This then becomes a convex optimization problem. We assume it's convex because if it's non-convex, finding the optimal solution can be non-trivial because it's really hard to guarantee Positive SemiDefiniteness of the Hessian in NP-hard cases. The learning problem then boils down to the following optimization algorithm:

$(\tilde{\theta}_1, \tilde{\theta}_2) = argmin_{\theta_1\theta_2} || Re(\theta_2, Co(Cr(\theta_1, x))) - x||^2$ where x is the original image, θ_1 and θ_2 are the parameters of ComCNN and RecCNN, respectively. Co(.) represents an image codec as described above.

III. Expected Outcome

For us to be able to measure the output, we'll test across various images and check metrics such as overall entropy, and compare with other baselines and existing techniques such as JPEG transformation and the paper in reference. This will be heavily tested depending upon the reliability of our network.

IV. Reference

1. Paper - https://arxiv.org/pdf/1708.00838v1.pdf
2. Kolmogorov Complexity - https://europepmc.org/article/PMC/PMC7516500
3. JPEG - https://papers.nips.cc/paper/2018/file/7af6266cc52234b5aa339b16695f7fc4-Paper.pdf
4. Deep Compression with Pruning & Trained Quantization- https://arxiv.org/abs/1510.00149