Visualizing and Understanding Convolutional Networks

Link

arXiv

Summary

- The authors propose a novel visualization technique for convolutional neural network. They use a multi-layered deconvolutional network to project feature activation back to the input pixel space. To examine a given unit's activation, they set all other activations in the layer to zero and pass the feature maps as input to the attached deconvnet layer.

- The reconstruction obtained for a single feature resembles a small portion of the original image where the pixels are weights according to their contribution toward that feature's activation. Since the models are trained discriminatively they implicitly show which part of the input image are discriminative for that category.

- They use an architecture similar to AlexNet for experimentation. The projections from each layer show the hierarchical nature of the features of the network. Layer 2 responds to corners and edges. Layer 3 has more complex invariances like capturing similar textures, text. Layer 4 captures more class specific features like dog faces, bird's legs etc. Layer 5 shows entire objects with significant pose variations, e.g., keyboards and dogs (AlexNet has 5 convolutional feature extraction layer).

- Lower layers of the model converge within a few epochs(their feature projection doesn't very much). However the upper layer features are fully developed after considerable number of epochs (40-50).

- Small translations and scaling have a dramatic effect on the first layer of the model but has lesser impact at the top feature layer, being quasi-linear for translation and scaling. This effect is measured by taking euclidean distance between feature vectors of original image and transformed image.

- Visualizing features revealed some problems with the original architecture. First layer filters were mix of extremely high and low frequency information with little coverage of the mid frequencies. Second layer visualization shows aliasing artifacts caused by large stride 4 used in 1st layer convolution. To remedy this they reduced first layer filter size from 11×11 to 7×7 and made the stride of convolution 2 instead of 4. This improved classification performance.

- They perform an occlusion sensitivity test where different portions of the input image is occluded with a grey square and monitoring the output of the classifier. This showed that the model was localizing objects correctly within the scene as the probability of the correct class drops significantly when the object is occluded.

- They were able to obtain much better result by using models pretrained on ImageNet dataset on much smaller Caltech-101, Caltech-256 and PASCAL 2012 dataset. For Caltech-101 dataset the accuracy drops from 86.5% to 46.5% if model is trained from scratch instead of pretraining on Imagenet. Similarly for Caltech-256 it drops from 74.2% to 38.8%. This suggests that features learned on ImageNet are general and suitable for other image classification tasks as well.