Analyzing The Performance of Multilayer Neural Networks for Object Recognition: Supplementary Material

Anonymous ECCV submission

Paper ID 1355

1 On Using Entropy For Measuring Change in Selectivity

1.1 Motivation

In answer to question - "What happens when a discriminatively pretrained network is fine-tuned" (sec 3 of the main paper) we have used entropy of filters and entropy of layers as a measure of how selective each layer is. The choice of entropy is motivated by: Entropy naturally measures how selective a particular feature is across different classes and not just individual classes. Consequently, entropy has been extensively used in a lot of computer vision applications based on decision trees such as [1] [2].

1.2 Method of Entropy Computation

We presented a natural way of computing entropy of each unit of every layer in sec 2.3 of the paper. We call this Label-Entropy. However, one might also compute entropy of units in other ways. We considered two more such methods as described below.

We define the entropy of a filter with respect to a given set of image-label paris in the following way. Each image, when passed through the convolutional neural network produces a $p \times p$ heatmap of filter responses. (eg p = 6, for a layer 5 filter). Further assume that there are C classes.

Label Entropy We vectorize the heatmap into a vector of scores of length p^2 and with each element of this vector we associate the class label of the image. Thus, for each image we have a score vector and a label vector of length p^2 each. Next, we concatenate score vectors and label vectors from N images into a giant score vector and a giant label vector of size Np^2 each. Now for every score threshold we consider all the labels which have an associated score $\geq$ to this threshold score. The entropy of this set of labels is the entropy of the filter at this threshold. As this threshold changes, entropy traces out a curve which we call as the entropy curve.

Weighted Label Entropy We vectorize the heatmap into a vector of scores of length p^2 and with each element of this vector we associate the class label of the image. Thus, for each image we have a score vector and a label vector of length p^2 each. Next, we concatenate score vectors and label vectors from N images into a giant score vector and a giant label vector of size Np^2 each. Now for every score threshold we consider all the labels which have an associated score $\geq$ to this threshold score. Next, we construct a C dimensional histogram, where each bin corresponds to a distinct class. The value of ith bin is the sum of scores of all examples which have label i and a score greater than threshold scores (Note: Since we are using the outputs of rectified linear units, all the scores are geq 0). The entropy of this histogram is the entropy of the filter at a given threshold. As this threshold changes, entropy traces out a curve which we call as the entropy curve.

Spatial-Max (spMax) Label Entropy From the heatmap, we chose the element which has the highet value and associate with it the label of the image class. Thus, for each image we have a score vector and a label vector of length 1 each. Next, we concatenate score vectors and label vectors from N images into a score vector and a label vector of size N each. Now for every score threshold we consider all the labels which have an associated score $\geq$ to this threshold score. The entropy of this set of labels is the entropy of the filter at this threshold. As this threshold changes, entropy traces out a curve which we call as the entropy curve.

1.3   Mean Cumulative Area Under Entropy Curve (MCAuE Index)

In order to compute entropy of the layer, we introduced the notions of Area Under the Entropy Curve and Mean Cumulative Area Under Entropy Curve (MCAuE) analogous to the notion of computing area under Mean Average-Precision ([3]) in sec 3 and fig 2 of the main paper. However due to space constraints we could only provide a breif defintion of MCAuE index. A detailed definition is as following:

For each filter in the layer we first calculate the area under entropy (AuE) curve. Next, filters are sorted according to their AuE. Then, we calculate the Cumulative Area under Entropy (CAuE) by computing the cumulative sum of sorted list of entropies of all filters of this layer. Finally, to normalize for the number of filters we divide the ith entry of the CAuE list by i. This results into a number which we call as the Mean Cumulative Area Under the Entropy Curve (MCAuE). In order to compare different layers, we compare MCAuE at 30 threshold points. The kth threshold point for layer l corresponds to MCAuE of top $N_l(k/30)$ filters, where N_l is number of filters in layer l.

1.4   Discussion

The change in MCAuE for different layers as measured using alternative definitions of entropy can be seen in fig 1 (Please see sec 3 of the main paper for

Table 1: This table lists percentage decrease in MCAuE as a result of finetuning when only 0.1, 0.25, 0.50 and 1.00 fraction of all the filters were used for computing MCAUE. A negative value indicated increase in entropy. Note that for all the metrics maximum decrease in entropy takes place while moving from layer 5 to layer 7.

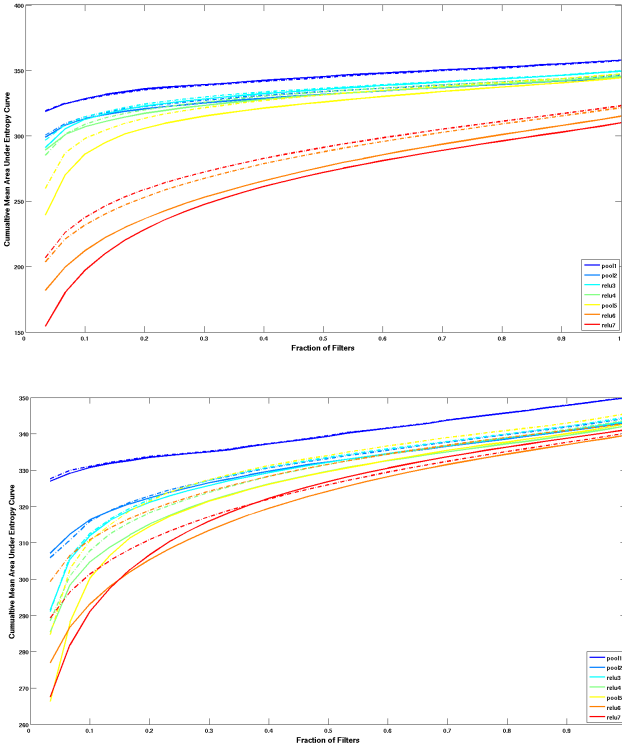| Layer	Label-Entropy				Weighted Label-Entropy				spMax Label-Entropy			
	0.1	0.25	0.5	1.0	0.1	0.25	0.5	1	0.1	0.25	0.5	1.0
pool-1	−0.02	−0.14	−0.19	−0.19	0.06	−0.13	−0.16	−0.16	0.19	0.10	0.07	0.04
pool-2	−0.71	−0.31	−0.14	0.01	0.41	0.53	0.58	0.57	−0.39	−0.03	0.11	0.23
relu-3	−1.14	−0.86	−0.67	−0.44	1.11	0.66	0.52	0.32	0.14	0.20	0.32	0.33
relu-4	−0.54	−0.31	−0.19	−0.05	−0.10	0.55	0.64	0.57	0.93	0.97	0.80	0.65
pool-5	0.97	0.55	0.43	0.36	5.84	3.53	2.66	1.85	4.87	3.05	2.31	1.62
relu-6	6.52	5.06	3.92	2.64	9.59	7.55	6.08	4.27	6.52	5.06	3.92	2.64
relu-7	5.17	2.66	1.33	0.44	20.58	14.75	11.12	7.78	5.17	2.66	1.33	0.44

Fig. 1: Mean Cumulative AuE plotted as fraction of filters for all layers of the Conv-Net. Dash-Dot Line: Alex-Net, Solid Line: Fine-Tuned Network. The top plot shows entropy calculated using Weighted Label-Entropy Method, whereas the bottom plot entropy calculated using spMax Label-Entropy Method.

more details on the procedure). A quantitative measure of change in entropy after finetuning is provided in table 1. The percentage change in MCAuE is calculated as:

$$\text{Percent Decrease} = 100 \times \frac{MCAuE_{untuned} - MCAuE_{fine}}{MCAuE_{untuned}} \tag{1}$$

where, $MCAuE_{fine}$ is for fine-tuned network and $MCAuE_{untuned}$ is for network trained on imagenet only.

Although, the most natural way to compute entropy is Label-Entropy as used by [2], [1], the alternative metrics also support the claim that most of the changes due to fine-tuning happen while moving from layers 5 to 7. At the same time, it is noteworthy, that for label-entropy pool-5 undergoes negligible change in entropy, whereas changes in pool-5 as measured by Weighted-Label Entropy and spMax-Label-Entropy are comparable to changes in relu-6 and relu-7.

2   On Grand-Mother Cells

3   Epoch Numbers

The Imagenet Training(Alex-Net) was performed using 310000 iterations, which corresponds to 33 epochs. (1 epoch is one pass over the entire training set.) For fine-tuning, we used a scheme closely following [4] which samples positives with a higher probability as compared to the negatives and hence reporting epoch numbers is not insightful. Please look into [4] for more details.

References

1. Geman, D., Amit, Y., Wilder, K.: Joint induction of shape features and tree classifiers. (1997)
2. Breiman, L.: Random forests. Mach. Learn. **45**(1) (October 2001) 5–32
3. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88**(2) (2010)
4. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. arXiv preprint arXiv:1311.2524 (2013)