50.039 – Theory and Practice of Deep Learning

Alex

Week 06: The Small Project

[The following notes are compiled from various sources such as textbooks, lecture materials, Web resources and are shared for academic purposes only, intended for use by students registered for a specific course. In the interest of brevity, every source is not cited. The compiler of these notes gratefully acknowledges all such sources.]

Submission due 21st of March 10pm

1 The small class project

What I want you to learn from it: Design a custom loss for a problem which is not completely standard.

Work with a custom loss on a bit more challenging vision dataset: Pascal VOC 2012 (its way less data than MIT places or LSUN, thats why I gave this to you!)

Single-label images from ImageNet

Multi-label images from Pascal VOC

img credit: HCP: A Flexible CNN Framework for Multi-label Image Classification, Yunchao Wei et al.

There are other nice vision datasets: MS COCO, MIT Places, LSUN Challenge (small number of classes) – but they all have much more images, and thus are way slower to train.

It has one important property: it is not a multiclass dataset, and in that sense way more realistic than imagenet

- each image can have multiple groundtruth labels, e.g. aeroplane and bird (and a car) present as labels for the same image.
- write a dataloader for pascal VOC train and val datasets, you can use the multilabel extractor code `vocparseclslabels.py` provided to avoid dealing with the xml files. Important: one image now is present in sets of multiple classes, so looping over classes of `self.images[class].append(filename)` is a bad idea for a dataset class, because if you would do so, then an image will appear multiple times then in the dataset.
- as model use one deep learning model, but with simultaneous outputs for all 20 classes. Transfer Learning is strongly suggested.
- come up with a proper loss for minimizing and for training of the network. note that an image can have multiple labels present in it. **Thus cross-entropy-loss over 20 classes, or any other multi-class loss, is not the right way to do here and will result in a large penalty.** Use a loss which can minimize 20 separate binary classifiers. How to design that ? It is easy if you think about it for 15 minutes.
- report the average precision measure (google for it, pascal voc has a matlab implementation, sci-kit learn surely has it too) on the validation set – for every class of the 20, and the mean average precision over all 20 classes. (Consider for yourself why accuracy is not an informative measure (e.g. by comparing accuracies vs average precisions).)
- for 5 random classes check visually the top-50 highest scoring images and the the top-50 lowest scoring images. To understand this: you have for every class a binary classifier. You can compute a score for every image for a given class. For a given class then you can sort images according to their scores for a given class.

  In the report show for 5 random classes the top-5 highest scored images, and the top-5 lowest scoring images (both for the same classes).
- why the top-50 highest scoring images for a given class looks so well when the ranking measure (average precision) is not perfect ? Compute for each of the 20 classes the accuracy of predictions in the upper tail, for 10 to 20 values of t from $t = 0$ if classification threshold is zero, or from $t = 0.5$ if classification threshold is 0.5 , until $t = \max_x f(x)$.

$$Tailacc(t) = \frac{1}{\sum_{i=1}^{n} I[f(x_i) > t]} \sum_{i=1}^{n} I[f(x_i) = y_i] I[f(x_i) > t], t > 0$$

Show in your final report a plot of $Tailacc(t)$ averaged over all 20 classes for 10 to 20 values of t as above. Note how the accuracy increases as we look at the more top-ranked results – this is the explanation.

The point here is to show you, that it is a matter of HCI how to deal with the errors of a DL system!

- write a report of 2 to 5 pages on what you did. Aim is so that others would be able to reproduce your results - it should contain what is needed to recode your results. Note down model, loss learning rate schemes, training procedures, **all relevant hyperparameters used in evaluation and the finally chosen hyperparameters** and so on. It does not need to be lengthy, just be self-containing. Short sentences are okay, prettyness of language does not matter. It is not an english language essay contest.

How good are you compared to the state of the art?

- `http://host.robots.ox.ac.uk:8080/leaderboard/displaylb.php?challengeid=11&compid=2` reports state of the art scores on the pascal VOC test set - this is not the validation set. Thus is it not 100% comparable.
- important: if you consider to submit a result to the pascal VOC test server, note that you can do only 2 submissions per week. Please refrain from spamming it – if you really want to submit, do it if your average precision on val is above 85 or so, and please submit max 2, but hide it from the leaderboard. Imagine if hundreds of deep learning courses would submit ... ?!
- You are not expect to reach the top-scores on this dataset. Not all problems are easy like MNIST. Another point to see that it is easy to get an okay score (see demonstrator) but at the same time many harder problems need rounds of iterative improvement. This kind of iterative improvement is what I would want to teach you in this class if I would have more time for it.

Takeaway for this lesson:

- think about what loss you really want to optimize for and to measure
- The loss matters as much as the model, if not more. The loss defines what your model will be good for.