14
Evaluating multiple ideas in parallel during error analysis

Your team has several ideas for improving the cat detector:

- Fix the problem of *dogs* being recognized as cats.
- Fix the problem of "*great cats*" (lions, panthers, etc.) being recognized as house cats (pets).
- Improve the system's performance on *blurry* images.
- …

You can efficiently evaluate all of these ideas in parallel. I usually create a spreadsheet, and fill it out while looking through ~100 misclassified dev set images, also jotting down comments that might help me remember specific examples. Illustrating this process with a small dev set of four examples, your spreadsheet might look like:

Image	Dog	Great cat	Blurry	Comments
1	✔			Unusual pitbull color
2			✔	
3		✔	✔	Lion; picture taken at zoo on rainy day
4		✔		Panther behind tree
% of total	25%	50%	50%	

Image #3 above had both the Great Cat and Blurry columns checked: it is possible for one example to be associated with multiple categories. This is why the percentages at the bottom don't necessarily add up to 100%.

Although I have described this process as first formulating the categories (Dog, Great cat, Blurry) and then going through the examples to categorize them, in practice, once you start looking through examples, you will likely be inspired to propose new error categories. For example, perhaps after going through a dozen images, you realize a lot of mistakes are on pictures that have been preprocessed by an Instagram filter. You can go back and add a new "Instagram" column to the spreadsheet. Manually looking at examples that the algorithm got wrong, and asking how/whether you as a human could have gotten the label right, will often inspire you to come up with new categories of errors and solutions.

The most helpful error categories will be ones that you have an idea for improving. For example, the Instagram category will be most helpful to add if you have an idea for "undoing" Instagram filters to recover the original image. But you don't have to restrict yourself only to error categories for which you already have an idea for improving; the goal of this process is to build your intuition about what areas are most promising to focus on.

Error analysis is an iterative process. You can even start off not having any categories in mind. Through looking at images, you might come up with a few ideas for categories of errors. Then after going through and manually categorizing some images, you might be inspired to create new categories, and so go back to re-examine the images in light of the new categories, and so on.

Suppose you finish carrying out error analysis on 100 dev set examples, and get this:

Image	Dog	Great cat	Blurry	Comments
1	✔			Usual pitbull color
2			✔	
3		✔	✔	Lion; picture taken at zoo on rainy day
4		✔		Panther behind tree
…	…	…	…	…
% of total	8%	43%	61%	

You now know that a project to address the Dog mistakes can eliminate at most 8% of the errors. Working on Great cat or Blurry image errors could help more. You might thus pick one of the two latter categories to work on. If your team has enough people to pursue multiple directions in parallel, you can also ask some engineers to work on Great cats, and others on Blurry images.

Error analysis does not result in a rigid mathematical formula that tells you what should be the highest priority task. You also have to take into account how much progress you expect to make on different categories, and the amount of work needed to tackle each one.