Carrying Out Error Analysis

<Look at dev examples to evaluate ideas>

90% accuracy
10% error

Should you try to make your cat classifier do better on dogs? ← focus on the dog problem?

Error analysis: → 5~10 min

- Get ~100 misabeled dev set examples. ↓5% 5/100
- Count up how many are dogs.

10% → 9.5% error (not effictive) that

↓50% 50/100 10% → 5% effictive

"ceiling"

what's in the best case
how well could working on the dog problem help you

"error analysis": effictive to deciede what is the most important thing / what to focus.

<Evaluate multiple ideas in parallel>

Ideas for cat detection:

- Fix picture or dogs being recognized as cats ←
- Fix great cats (lions, panthers, etc..) being misrecognized
- Improve performance on blurry images.

draw the table

100

Image	Dog	Great Cats	Blurry	Instagram	Comments
1	✓			✓	Pitbull
2			✓	✓	
3		✓	✓		Rainy day at zoo
⋮	⋮	⋮	⋮		
% of total	8%	43%	61%	12%	

estimate of how worthwhile it might be to work on each of these different categories of errors.

ceiling of data is much higher.

Cleaning up Incorrectly labeled data

Incorrectly labeled data → fix it!

<Incorrectly labeled examples>

X	cat	dog	cat	cat	dog	dog	cat
y	1	0	1	1	0	1	1

incorrect label.

Should be 0 but get wrong.

Training set.

DL Algorithms are quite robust to random errors in the training set.

Systematic errors: less robust.

<Error Analysis>

Image	Dog	Great Cat	Blurry	Incorrectly labeled	Comments
...					
98				✓	Labeler missed cat in background ←
99		✓			
100				✓	Drawing of a cat; Not a real cat. ←
% of total	8%	43%	61%	6%	

Overall dev set error	10%
Errors due incorrect labels	0.6% small..
Errors due to other causes	9.4% ←

focus on fixing!

2% 0.6% 1.4%

20% higher fraction → fix ok

fixing it is not the most incorrect right now (0.6% - very small)

Goal of dev set is to help you select between two classifiers A & B.

if it makes a significant difference to your ability to evaluate algorithms on your dev set, then go ahead and spend the time to fix incorrect labels.

doesn't make a significant difference → not be the best use of your time.

<correcting incorrect dev/test set examples>

- Apply same process to your dev and test sets to make sure they continue to come from the same distribution.

- Consider examining examples your algorithm got right as well as ones it got wrong. 2%
  98%
  ↳ doesn't always done. Just something to consider

- Train and dev/test data may now come from slightly different distributions.

Build First System Quickly, Then Iterate

- Set up dev/test set and metric  set up a target
- build initial system quickly
- use bias/variance analysis & Error analysis to prioritize next steps.