§ ML Strategy

1. Chain of assumptions in ML

Fit training set well on cost function → bigger network / Adam
↓
... dev ...... → Regularization / Bigger train set
↓
... test set ... → Bigger dev set
↓
performs well in a real world → change dev set / cost function

2.

- precision: of examples recognized as cat, what % actually are cats
- Recall: what % of actual cats are correctly recognized

F1 = "average" of P and R

$$\to \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

3. Size of data (Big data):

- Train: 98%
- Test: 1%
- Dev: 1%

→ enough

Set test set to be big enough to give high confidence in the overall performance of your system.

4. doing well on metric + dev / test set

↓ × correspond to do well on application

→ change metric / dev / test

5. ML worse than humans:

① get labeled data from humans

② gain insight from manual error analysis: why did a person get this right?

③ better analysis of bias / variance

6.

human error	1%	7.5%
training error	8%	8%
dev error	10%	10%

avoidable bias   focus on bias   focus on variance

variance

7. Two fundamental assumptions of supervised learning

① You can fit the training set pretty well.

② The training set performance generalizes pretty well to the dev/test set.

8. Human level

avoidable bias → Train bigger model / Train longer / better optimization algorithms / NN architecture / hyperparameters search

Training error

variance → More data / Regulation / NN architecture / hyper parameters search

Dev error

9. Clean up incorrectly label data

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

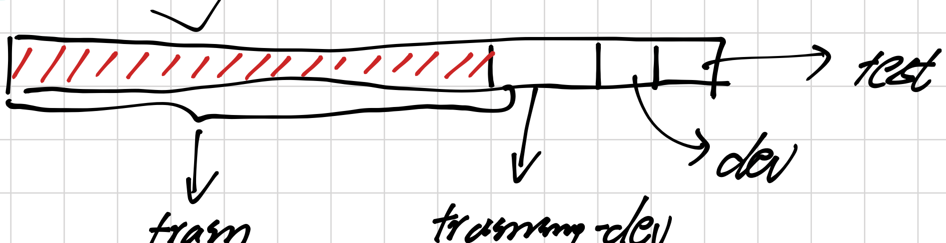
Correcting incorrect dev/test set examples:

① Apply same process to your dev and test sets to make sure they continue to come from the same distribution.

② Consider examining examples your algorithm got right as well as ones it got wrong.

③ Train and dev/test data may now come from slightly different distributions.

10. Training-dev set: Same distribution as training set, but not used for training



	①	②
Training error	1%	1%
Train-dev error	9%	1.5%
Dev error	10%	10%

① variance

② data mismatch

Human error	0%	
↕ available bias		available bias
Training error	10%	10%
↕ variance		
Training-dev error	11%	11%
↕ data mismatch		data mismatch
Dev error	12%	20%

⇒ general formulation

	general speech recognition	rear view mirror speech data
human level	Human level 4%	
	↕ available bias	
error on examples trained on	Training error 7%	
	↕ variance	
error on examples not trained on	Training dev error 10% ⟷ data mismatch	Dev/test error 6%

11. data mismatch

① Carry out manual error analysis to try to understand different between training and dev/test sets.

② Make training data more similar, or collect more data similar to dev/test set.

12. Transfer Learning

$A \to B$

① A and B have the same input X

② You have a lot more data for A than B

③ Low level features from A could be helpful for learning B.

13. Multi-task Learning

① Training on a set of tasks that could benefit from having shared low-level features

② Usually: Amount of data you have for each task is quite similar.

③ Can train a big enough NN to do well on all tasks.

14. End-to-end deep learning

Pros:

1. let data speak
2. less hand-designing of components needed

Cons:

1. May need large amount of data
2. Excludes potentially useful hand-designed components