PAGE
DATE

(course3)
- Structuring Machine learning projects.
(week1)
→ Why ML strategy. intro ✓
→ Orthogonalization
. it is to have clear eyed about what to tune inorder to try to achieve one effect
. in supervised system we usually need to tune the knobs of system to make sure that four things hold true
① make sure that that you're at least doing well on trainset
② leads to doing well on the devset
③ " " " " " testset
④ " " " " on tests set on that cost fun.
① . If the algo. not fitting the training set well on the cost fun. so you might train bigger network (or) switch to better optimization algorithm
② . If alg. is not doing well on devset then regularization is changed (or) getting bigger training set
③ . then we want bigger devset
④ . we need to go back & change cost fun. (or) devset.

→ Setting up your goal.
→ single number evaluation metric
. whether tuning hyperparameters (or) trying out different ideas for learning algorithms, or just trying out diff options for ML system, you'll find the progress much faster if having single number evaluation metric
. refers to metric that summarizes the model performance

PAGE
DATE

- from ways to evaluate the performance of classifiers is
  → precision : of examples that classifier recognize as cats what % actually are cats.
  → recall : of all the images that are really cats, what % were correctly recognized by your classifier.

	precision	recall
A	95%	90%
B	98%	85%

- if classifier A does better on recall & classifier B does better on precision, then you're not sure which classifier is better
- new evaluation metric that combines precision & recall is called f1 score) → avg of precison & recall

$$\left(\frac{2}{\frac{1}{P}+\frac{1}{R}}\right) \rightarrow \text{Harmonic mean}$$

→ satisficing & optimizing metric ⊛

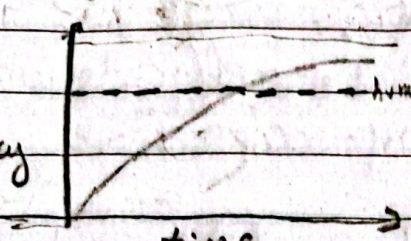
→ train / Dev / test Distribution
- Dev & tes should come from same distribution or same source & they should be randomly shuffelled into dev/test

→ size of Dev & test set
- set test set to be big enough to give high confidence in the overall performance of system

PAGE
DATE

→ Comparing to human level performance

→ why human level performance



• progress relatively fast as you approach human performance after surpassing human then prog. accuracy is slow & the performance never passes Theoretical limit called The Bayesian optimal error

• as long as M/L is worse Then humans you can:
① get labeled data from humans ② gain insights from manual error analysis
③ Better analysis of Bias/Variance

• The distance between Bayesian error & the training error called The avoidable bias & the distance between training error & Deverror is Variance.

Humans ≃ bayesian 7.5%
train error 8%
dev error 10%
Variance 2%

*(Course 3)(week2) error analysis

→ Carrying out error analysis

• if learning algorithm is not up to The level of human, then directly identifying the mistakes The algorithm is making will give you insights on what to do next & this process is called error analysis

→ Cleaning up incorrectly labelled Data

- If error have significant impact on evaluating alg. using development set then we should spend time fixing mislabeling

→ Build your first system quickly, then iterate

① setup dev/test set & metric

② Build initial system quickly

③ use Bias/Variance analysis & error analysis to prioritize next step

→ mismatched training & dev/test data

→ Training & testing on diff. distributions

Data from web: 200 000

Data from mobile app.: 10 000

→ Shuffle

X option1

train	dev	test
205,000	2,500	2500

} with shuffle

option 2

web & mob	dev	test
205000	mob: 2,5	2,5

} without

→ Bias & Variance with mismatch data distributions

- we define training-dev set: same distribution as training set but not used for training.

train	train-dev	dev	test

Training error	1%	
Train-dev error	9%	} variance
dev error	10%	} data mismatch
test error	12%	} degree of overfitting