Course-3 Structuring ML Projects

Week-1 ML project might not work well in such a case, we have following options

Collect more data
" " diverse training set
Train algorithm longer w/ gradient descent
Try Adam instead of " "
Try bigger n/w
" smaller "
" dropout
Add L_2 regularization
Network architecture
Activat^n f^n
#hidden units
...

In this course, strategies will be taught on how to analyze issues of a n/w
The process to tune, in order to achieve one effect, is known as orthogonalizat^n.

Using a Single Number Evaluation metric

Classifier	Precision	Recall
A	95%	90%
B	98%	85%

ecision = Of samples recognized as cats, what % actually are cats?
ecall = What % of actual cats are correctly recognized

There is certain trade-off going on between precision and Recall, thus, we take a single evaluation term called F1 score. It is harmonic mean of Precision & Recall

$$\text{F1score} = \frac{2}{\frac{1}{P} + \frac{1}{R}}$$

Better classifier ⟶ ↑ F1 score

Similarly for cats ~~dogs~~ cat lovers in 4 geographies. Diff errors by classifiers

Algo	US	China	India	Other	Average
A	3%	7	5	9%	6%
B	5%	6	5	10%	6.5%
C	2%	3	4	5%	3.5%
D	5%	8	7	2%	5.25%
E	4%	5	2	4%	3.75%
F	7%	11	8	12%	9.5%

We choose the one whose avg. error is least hence instead of evaluating the whole grid, we just need evaluate a single number matrix.

It is not always easy to make a ~~simple~~ single matrix each time.

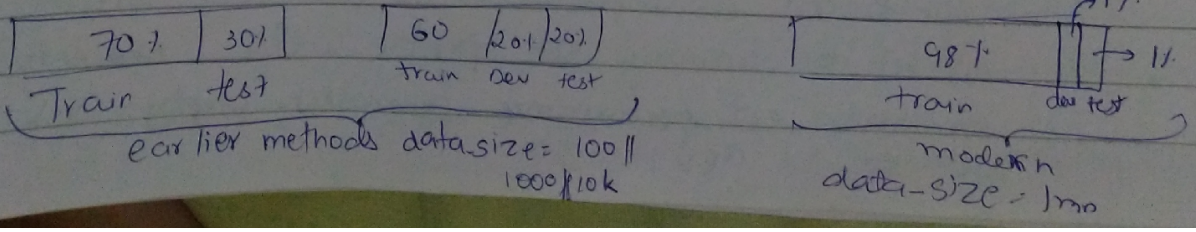
Classifier	Accuracy	Running Time
A	90%	80ms
B	92%	95ms
C	95%	1.50ms

Instead of using some random formula to select the best classifier, what we do is

maximize accuracy
suffice to running Time ≤ 100ms

In such a case, accuracy becomes optimizing matrix, and Running Time becomes satisficing matrix

→ Dev and test set should come from same distribution. But, how large should they be.



Many a times, it may happen that the classifier has less performance but does quite well in real life scenarios.

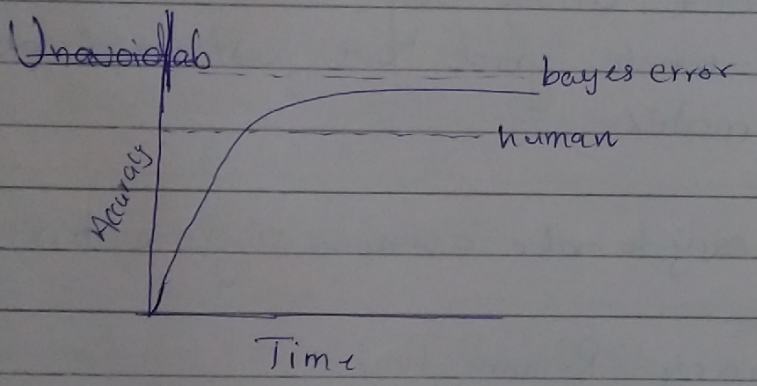
For e.g. cat images with high definit^n are used to train

In that case, ~~cat~~ classifier A → 3% error

classifier B → 5% "

But it turns out that when low quality images are used, B works better.

Hence B is chosen



No algorithm can surpass bayes error

For a ~~hoo~~ classifier which ~~coho~~ identify cats:

	A		A		
~~Trainii~~ Humans	1%	↓7%	7.5%	↓0.5%	↕ Avoidable Bias
Training error	8%	↓2%	8%	↓2%	↕
Dev error	10%		10%		↕ Variance

Focus on Bias | Focus on variance

We consider human level error as a proxy for Bayes error

{My thoughts: Focus the one whose error is less.}

Two functional assumptions of supervised learning

1. You can fit the training set pretty well ~~min~~ ∿ Avoidable bias
2. The training set performance generalizes pretty well to the dev/test set } ∿ Variance

Reducing avoidable bias
- Train bigger model
- Train longer/better optimizat^{n} algos (eg: RMSprop, momentum, Adam)
- NN archi./hyperparameter search
  - ↳ RNN, CNN (model change)
  - #hidden units
  - activation change

Reducing Variance
- More data
- Regularization (eg: L2, dropouts, data augmentat^{n})
- NN architecture / hyperparameters search
  (same as above)