C3w1 ML STRATEGY

Why ML stategy:-

There are couple ways to improve machine Learning model:-

① Collect more data
② Collect more diverse training set.
③ train algorithm longer with gradient descent.
④ Try Adam instead of gradient descent.
⑤ Try bigger network
⑥ Try smaller network
⑦ Try dropout
⑧ Try L_2 regularization
⑨ Change network architecture:-
   (i) activation function
   (ii) #hidden unit.

Orthogonalization :-

Let, a car have two tuning parameter ① steering ② speed. if we had single switch or controller for both of these then it would be harder to control the car but if the controller are separate it is easy to control the car. the term orthogonal means putting each controll in 90° with each other.

Chain of assumptions in ML:-

Fit train well on cost function
- bigger network.
→ tune hyperparameter
- Optimize

→ Fit dev set well on cost function
- regularization.
- Bigger test set

→ Fit test set well on cost function
- Bigger dev set.

→ Performs well in real world.
- Change cost function.
- change dev set.

Setting up Goal:-

single number evaluation metric

We have two classifier that gives us following result:-

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

Precision → Of the example recognized as cat, what percentages actually are cat.

Recall → what % of cat are correctly recognized

Classifier A works better on Recall. And Classifier B works better on precision. So if we want to choose best among those two we would be confused. so we should use a single metric to measure the performance of a model. Such as F1 score

F1 score = "Average of precision & recall".

$$= \frac{2}{\frac{1}{P} + \frac{1}{R}}$$ Harmonic mean

	F1 score
A	92.4%
B	91.0%

A is better.

Satisficing & Optimizing metric:-

A cat classifier.

classifier	accuracy	running time
A	90%	80ms
B	92%	95ms
C	95%	1500ms

Optimizing :- Our goal is optimize this as much as possible.

satisficing :- Our goal is just to pass the threshold. if it passes the threshold then we don't care about it's value.

if we have N metric : 1 optimizing
N-1 satisficing

Train/dev/test distribution:-

Dev & test data should always come from the same distribution

<u>size of dev & test:-</u>

old way of splitting data:-

70% train | 30% test

60% train | 20% dev | 20% test

} m < 1M.

current → 98% T | 1 D | 1 T → m ≥ 1M

<u>Change dev/test set metric :-</u>

Lets we have 2 Algorithm for cat classifier.

Algorithm A :- 3% error.

Algorithm B :- 5% error.

it seems like algorithm A works better but it label more pornographic image as cat. And we don't want to show porn to the user. so in this case Algorithm B is better.

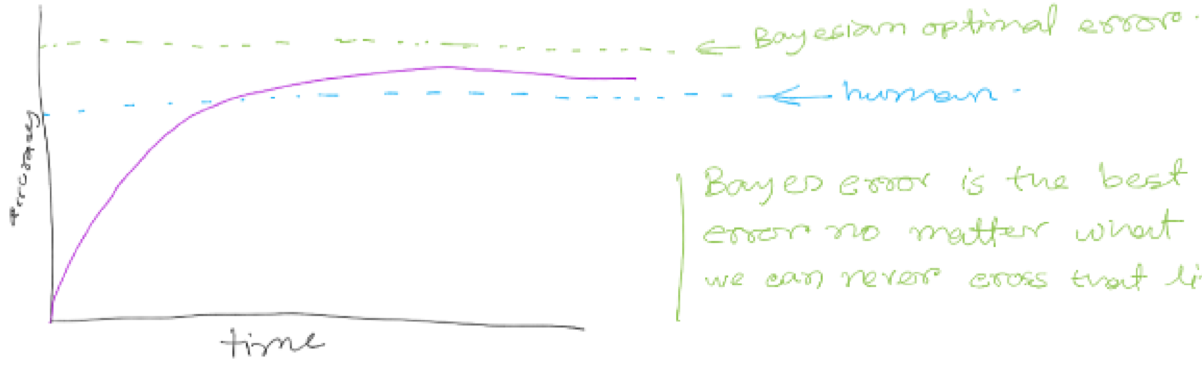
we can modify cost function for this problem:-

We can redefine cost function for this problem.

$$\text{Error:- } \frac{1}{\sum w^{(i)}} \cancel{\frac{1}{m_{dev}}} \sum_{i=1}^{m_{dev}} w_i \, \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

$$\rightarrow w_i = \begin{cases} 1 & \text{if } x^{(i)} \text{ is non porn.} \\ 10 & \text{if } x^{(i)} \text{ is porn} \end{cases}$$

Comparing to human level performance:-



Bayes error is the best possible error no matter what we do we can never cross that limit.

Avoidable bias:-

	Case A	Case B
Human error	1%	7.5% ↕ avoidable bias (0.5)
training error	8%	8%
Dev error	10%	10%
	focus on bias	focus on variance

in computer vision problem human error can be used as a proxy of Bayes error. Cause humans are pretty good at recognizing object.

Reducing bias & variance:-

Human Level.

↑

to reduce bias →

- train bigger model.
- train longer.
- Better optimization algorithm...

Better optimization algorithm.
Change NN architecture.
Hyper-parameter search.

Train error.

to reduce variance →

More data
Regularization.
Change NN architecture
Hyperparameter search.

Dev error.