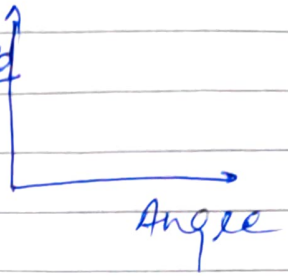
Camlin | Page
Date / /

STRUCTURE OF ML PROJECTS :-

Week - 01

Camlin | Page
Date / /

Orthogonalization :-



Chain of Assumption of ML

1. Fit training set well on cost function [BiSseoNet ADAM]
2. Fit dev set well on cost-function [Regularization Dropout] BiSseo train set
3. Fit test set well on cost-function [BiSseo Dev Set]
4. Perform well in real-world. [change Devset cost function]

$$\frac{x_1 + x_2 - x_1}{2}$$

$$\frac{x_1 + x_2 - x_1}{2}$$

$$\frac{x_2}{2}$$

Single Number Evaluation Metric

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

Combine them by F1 Score = $\frac{2PR}{P+R}$

Dev set + Single Real Number Evaluate Metric

Algo	US	China	India	other	Avg
A	3%	7%	8%	6%	6%
B	5%	6%	5	10	6.75

Take AVG to achieve Single Number Evaluation Metric

Classifier	Acc	Run Time
A	90%	80 ms
B	92%	95 ms
C	95%	1500 ms

we also care for Run time :-

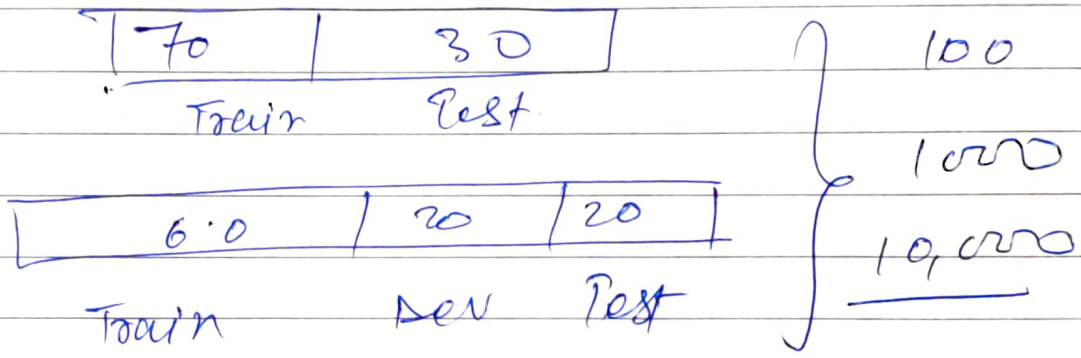
Choose classifier & MAX ACCURACY

Subject to Running Time $\leq$ 100 ms

Acc :- Optimizing / RunTim : Satisfying Metrix

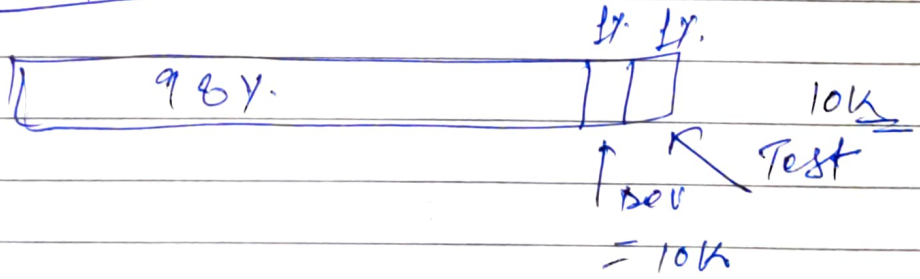
In general if N metric 1 optimizing & N-1 will be satisfying :-

Same Distribution for Dev / Test test



If we have 1 million train example :-

1,000,000 —



If we huge ~~Too~~ Large data-set then use much large ~~data~~ ~~Set~~ sub set can be used for TRAIN!

Improving Model Performance

Two fundamental assumptions of supervised learning

1. You can fit the training set pretty well that means low avoidable bias

2. The train'n set performance should be generalize pretty well to the dev/test set. that means low variance

- Avoidable Bias
  - Train Bigger Model
  - Train longe Better Optimization ALGO
  - Hyperparam Search
- Variance
  - More Data
  - Regularization - L2 / Drop out
  - Data - Augmentation

10,000,000 (10 million, y = 0 / 1)
(NoBird / Bird)

Metric Success :- 1. High Accuracy 2. Runtime less
3. Less memory.

Carrying Out Error Analysis

Ex. A classifier to classify CAT, having 90% accuracy 10% error. How to do error analysis:

1. Take 100 misslabeled example from DEV SET

2. Count how many dogs has been misclassified. Say it turns out 5% are DOG

3. So even if do very good on dog problem so my classifier accuracy will improve by 0.5% i.e error will improve by 0.5%. i.e 9.5% now.

4. Evaluate multiple Ideas in parallel

Ideas for cat detection:-

1. Fix pictures of dogs being recognized as cats.

2. Fix great cats (lions, panther etc) being misrecognized.

3. Improve performance on blurry images.

Image	Dog	Greatcat	Blur	Wrong Lbl		Comments
1				✓		Pitbull
2						
3				✓		
⋮				⋮		
100						Rain in zoo.
% of Total	8%	43%	61%	6%		

Cleaning Up Incorrect Label :- Can be dealas on. more column here. It's OK leave it If % of incorrect label is not more than RANDOM.

Over All Dev Set Error : 10%.

Error Due to Incorrect Lbl : 0.6%

Error Due to other cause : 9.4%.

< Dev / Test > must from same distribution

< Train > // < Dev / Test > may now come from slightly different distribution.

(X) Build Your First System Quickly then Iterate.

Mismatched Train // Dev/Test

Training & Testing on different DIST

Cat app example

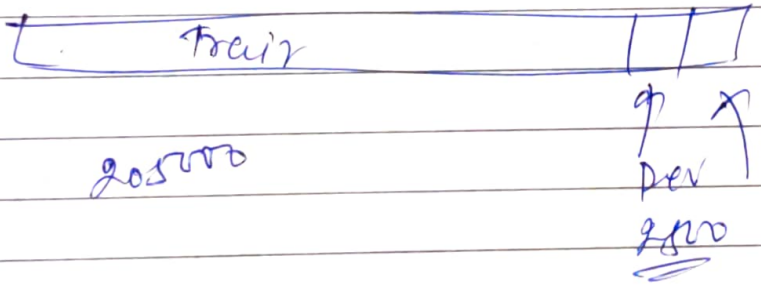
Train data from Web [High Res / Shalpele]
↳ 200,000

Test data from Mobile [Blur, shake etc]
10,000

Option 1:

200,000 // 1000 ←

(210,000) M/S & Shuffle



205000

2500 2500

Advantage: Train / Dev / Test come from same dist.

Dis Advatage: Dev Set (2500 approx

NOT GOOD ←

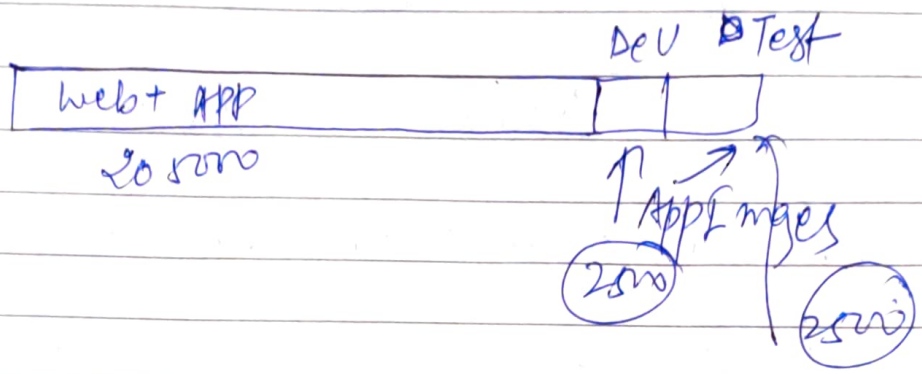
2381	119
web	Mobile

we are saying to do GOOD web images. That is not Moto.

OPTION 1 & NOT OK

Camlin Page
Date / /

Option 2:



You are targetting what we really care about.

Bias & Variance with mismatched data distributions

Cat classifier: Assume human get $\approx$ 0% error.

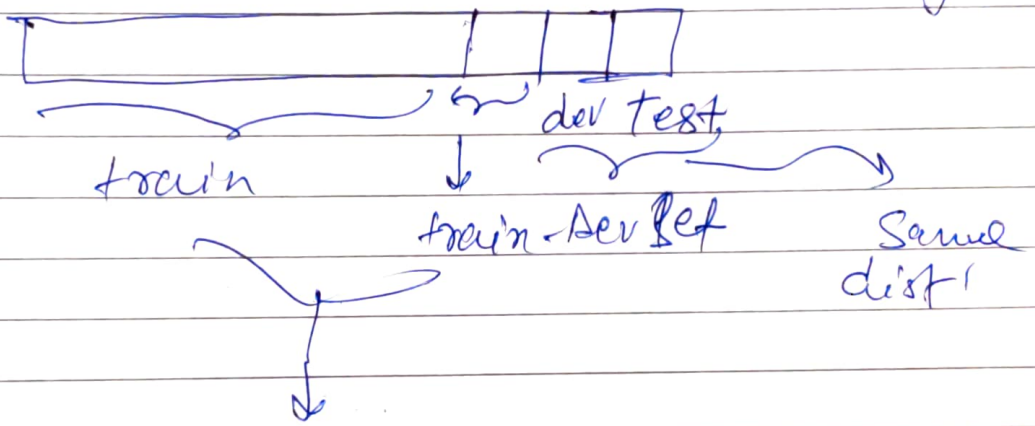
Train Error :- 1% } if train
Dev Error :- 10% } Dev have same distribution I will say there is a variance problem.

It is not generalizing well.

But if Dev & Train is from "DIFFERENT DISTRIBUTION"

So we coin a new term

Training-Dev Set :- Same dist as train but not used for training.



Train . 1%.
Train-Dev 9%.
Dev Error

So conclusion is we have VARIANCE problem.

Train 1%.
TrainDev 1.5%.
Dev Err 10%.

Data miss match problem.

Camlin Page
Date / /

Human Err ≈ 0%. ↑ → Avoidable
Train Err ≈ 10%. ↕ Bias problem
Train-Dev Err ≈ 11%.
Dev Err ≈ 12%. ↓

Human Err = 0%. ↑ → Avoid Bias problem as well as Data
Train Err = 10%. ↕ MISS MATCH"
Train Dev Err = 11%.
Dev Err = 20%. ↓

Human level Err ↑ 4%
Avoidable Bias
Train Set Err ↓ 7%
Variance
Train-Dev Err ↕ 10%.
Dev Err ↕ 12%. → Data Miss match
Test Err ↕ 12% → Degree of overfitting.

If test Err higher than maybe we have to find Bigger "Dev. Set".

Adressing Data Miss Match

⇒ Carry out manual error analysis to try to understand difference b/w training Dev / Test sets.

⇒ Make training data more similar

=)))) + Car Noise = Synthesized in-car studio.

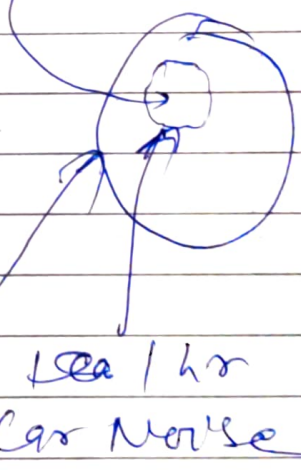
The quick Brow fox Jump ores the lazy dog

↓

1 hrs

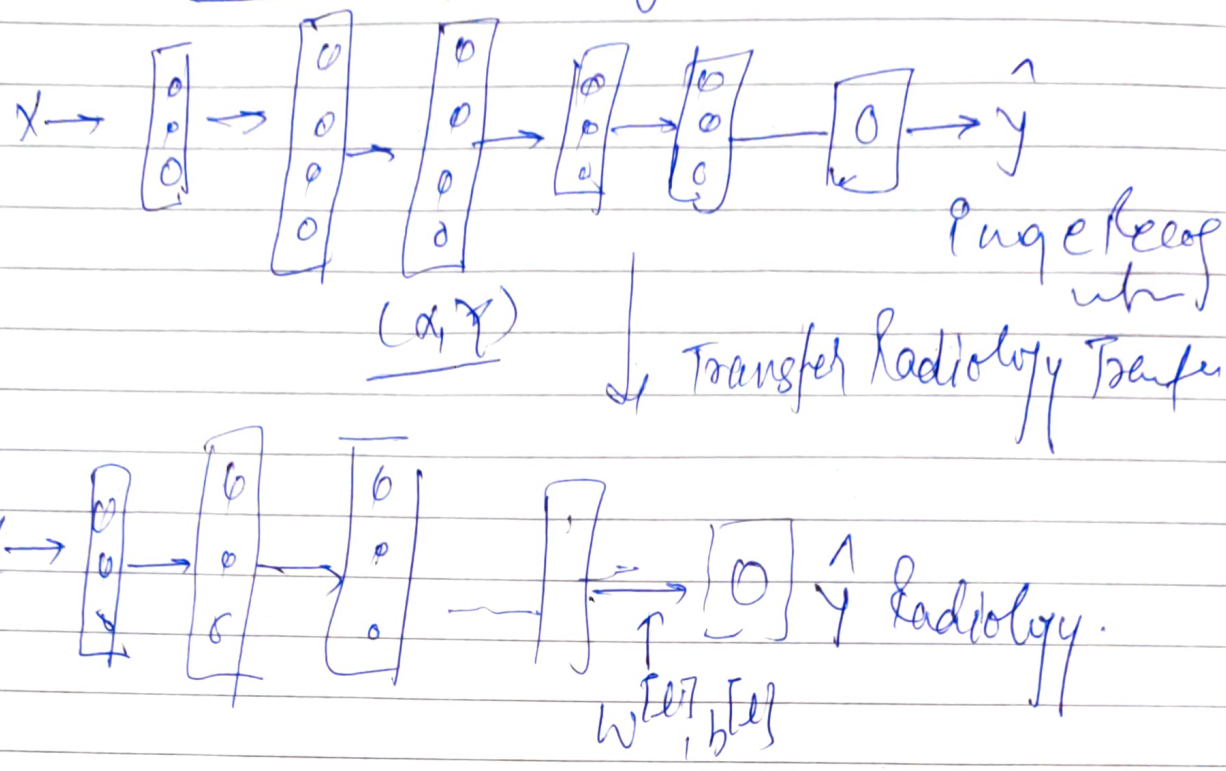
10,000 hrs

So we can overfit to 1 hr of car noise



Set of all audio of a car.

Transfer Learning



1,000,000 for cat
100 Radiology image } good transfer case.

Task A → Task B

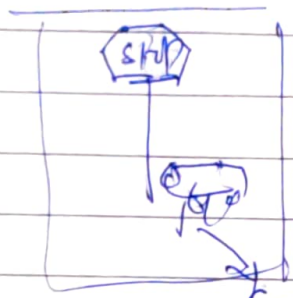
* Task A & B have same input x. Has
* You have lot more data for A as B.
* Low level feature from task A usefull for task B.

Camlin | Page
Date / /

Multi Task Learning

Detect Pedestrian, Car, Sign, Trafficlight

$x^{(i)}$

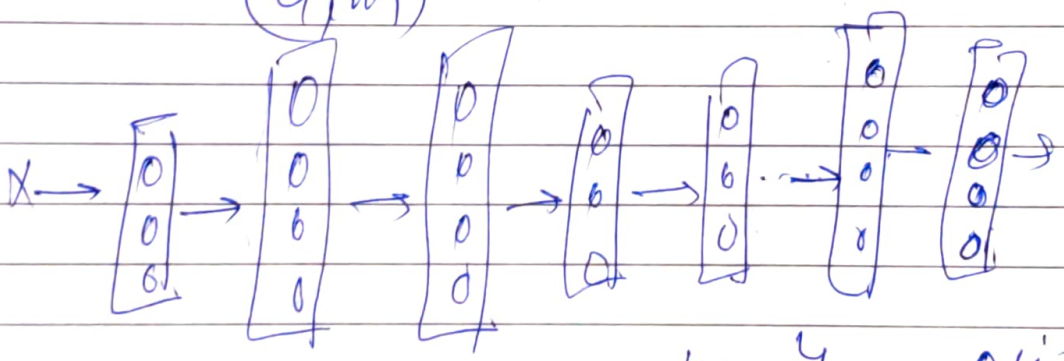


$x^{(i)} \longrightarrow \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$ (4,1)

$y^{(i)}$

$y = [\, y^{(1)}\ y^{(2)} \cdots y^{(m)} \,]$

(4,m)



Loss $\hat{y}^{(i)}$ (4,1)

$$\frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{4} \mathcal{L}(\hat{y}_j^{(i)}, y_j^{(i)})$$

$\mathcal{L}$ → logistic loss

$-y_j^{(i)} \log \hat{y}_j^{(i)}$

$-(1-y_j^{(i)}) \log(1-\hat{y}_j^{(i)})$

Unlike Softmax Regression one image can have multiple labels.

When multi-task make sense?-

1. Training on a set of tasks that could benefit from having shared lower-level feature.

2. Usually amount of data we use should be quite similar.

A_1, 1000
A_2, 1000
A_3, 1000
A_{100}, 1000

3. Can train a big enough neural-network to do well on all the tasks.

2 End-to-End Deep Learning

When to use deep learning?

Pros:- 1. Let the data speak $X \rightarrow Y$

2. Less hand design hand-components.

Cons:- 1. Need large amount of data

2. Excludes potentially useful hand-design system.