Orthogonalization: eg. when we tune one parameter to improve training set performance, we have the performance on dev, test or real world will be affected as little as possible

chain of assumption in ML

fit well on training set { bigger network; Adam
↓
fit well on dev set { Regularization; bigger train set
↓
fit well on test set { bigger dev set
↓
fit well on real world { change dev/test set or cost function

we try not to use early stopping because it will affect the performance on training and dev set at the same time, thus violate the principle of orthogonalization.

Evaluation

- precision : how many selected element are relevant
- recall : how many relevant element are selected
- F1 score = $\frac{2}{\frac{1}{P}+\frac{1}{R}}$

Selecting model with multi evaluation metric

eg.

classifier	accuracy	running time
A		
B		
C		

$\max_{A,B,C}$ accuracy

s.t. running time < 100s

Selecting dev/test

① similar to future data

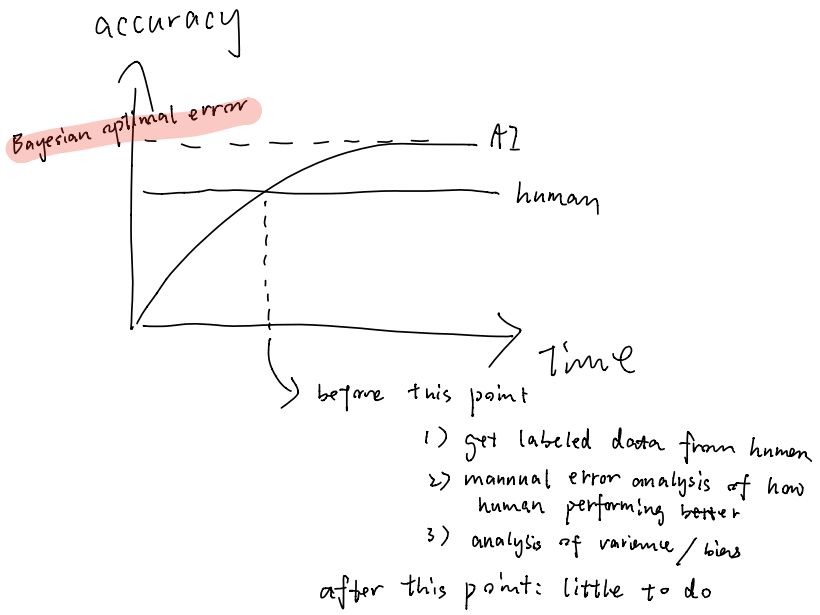
② dev, test should have same distribution

③ train/dev/test = 98%/1%/1% if the dataset is big enough

if current evaluation cannot capture the actual preference of algorithm

1) define a new evaluation metric

2) change test/dev set



eg.

human error (bayes)	1%	7.5%
training error	8% (Focus)	8%
Dev error	10%	10% Focus

AI surpass human with structured data task
eg. online advertising, product recommendation ...

but AI find it hard to surpass human in
natural perceptual task (es. NLP, CV)

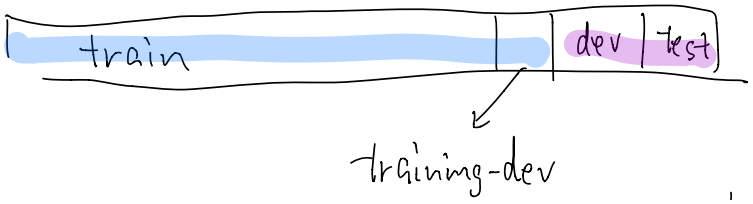
Error Analysis: whether a direct is worth working on

eg. cat detector

among 100 misclassified

No.	dog	big cat	blurry	...
1	✓			
2		✓		
i			✓	
100				
total				

Mismatch of distributing between train and dev/test



same distribution | same dist

bayesian	error	4 %
training	error	7 %
training-dev	error	10 %
Dev	error	12 %
Test	error	12 %

4 %: avoidable bias
7 %: variance (overfitting to train)
10 %: data mismatch
12 %: overfitting to dev

solution

1. manunal understand the problem
2. collect/make train data similar to dev/test
   eg. audio synthesis
   conversation + car noise

Transfer Learning

eg1. image recognition → radiology diagnosis
eg2. voice recognition → trigger word detection

条件 1. task A and task B have similar input
2. task A has much more data than task B
3. low level feature from A can help B

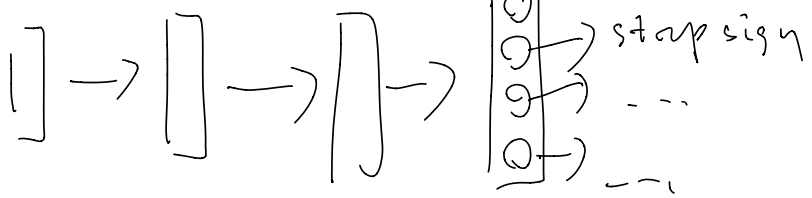
step 1. pre-training
先训练一个 image recognition 的 NN

Step 2. fix parameter → just change $w^{[L]}, b^{[L]}$

Fine-tuning

feed the model with radiology labeled data

Multi-task learning



$$J = \sum_{j=1}^{m} \sum_{i=1}^{4} L(y_i^{(j)}, \hat{y}_i^{(j)})$$

train 4 separate NN