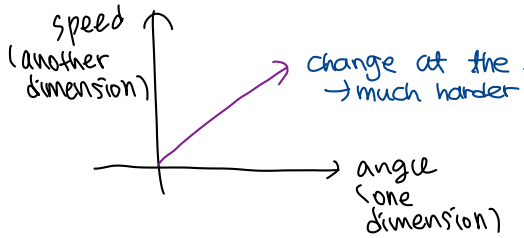
Orthogonalization

<TV / Car Example>

Q:
knobs → change



change at the same time
→ much harder to set!

in theory, by turning these two knobs
→ get your car to steer the angle and the speed you want much harder
if you just have one single control for controlling a steering angle

<Chain of assumptions in ML>

Fit training set well on cost function
(≈human level performance)
performance on the training set need to pass some acceptability affection.

knob → bigger network
atom optimization
...

early stocking
difficult.

Fit dev set well on cost function.

knob
regulazation
bigger train set...
different knob with training set

Fit test set well on cost function.

knob
bigger dept set...

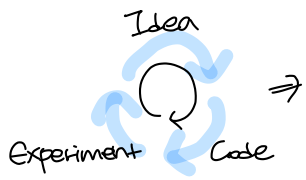
Performs well in real world
(ex happy cat picture app users.)

knob
Change dev set or cost function

either your dev set distribution isn't set correctly or your cost function isn't measurin

Single Number Evaluation Metric

<Using a single number evaluation metric>



ML → empiricial process

Classifier	Precision	Recall	F1 Score
A	95%	90%	92.4%
B	98%	85%	91.0%

trade-off

new evaluation
→ combines precision and recall.

Precision: of examples recognized as cat, what percentage actually are cats?

Recall: what % of actual cats are recognized

not sure which crossfire is better.

F1 score = "Average" of precision P and recall R

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

Dev set + Single number evaluation metric
real
⇒ Speed up iterating.

<Another Example>

Algorithm	US	China	India	Other	Average
A	3%	7%	5%	9%	6%
B	5%	6%	5%	10%	6.5%
					3.5%
					⋮

quickly tell

Single number evaluation metric can really improve your efficiency.

testing a lot of different compilers
→ difficult to look at all these numbers and quickly pick one.

Satisficing and Optimizing metrics.

more important: optimizing
less : satisficing

<Another Cat Classification Example>

optimizing metric — you also care about it..

Classifier	Accuracy	Running Time
A	90%	80 ms
B	92%	95 ms
C	95%	1500 ms

Satisficing metric

Cost = accuracy − 0.5 × running time

maximize accuracy
subject to running time ≤ 100 ms.

be less than 100 ms and beyond that you don't really care / don't care that much.

N metrix: 1 optimizing
N−1 Satisficing

ex) Wakewords / Trigger words — Alexa
OK Google
Hey Siri...

accuracy
false positive

maximize accuracy — optimizing metric
s.t. ≤ 1 false positive — satisficing metric
every 24 hours.

Train / dev / Test Distribution

how you set up your depth and test set).

<Cat classification dev/test sets>

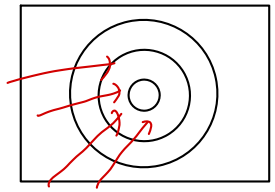
development set, hold out cross validation

Regins:
- US
- UK
- Other Europe
- South America
- India
- China
- Other Asia
- Australia

very bad idea → dev & test come from very different distributions.

Dev ←
very different
Test ←

not giving you a good performance on the test set (different location)



closer and closer..

dev set + evalution metric : tell you a goal

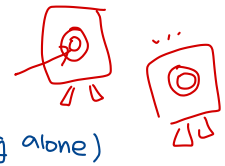
take all these data preventing the shuffle data into the dev/test set

Randomly shuffle into dev/test

<Time Story (details changed)>

Optimizing on dev set on loan approvals for medium income zip codes

X (about loan application)
↓ predict
Y (whether or not they'll be paying alone)
(repay loan?)



Tested on low income zip codes didn't work at all.

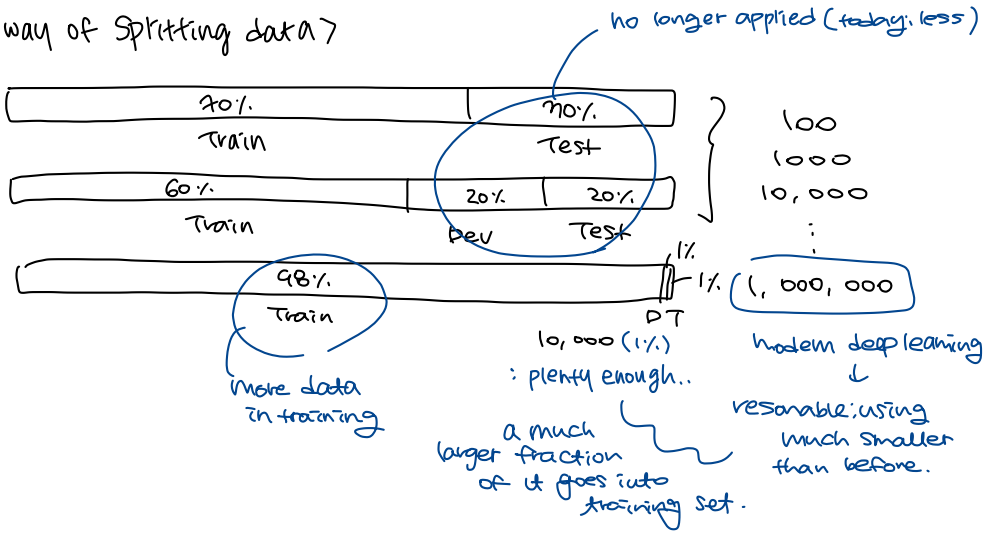
<Guideline 7>

Same distribution.

Choose a dev set and test set to reflect data you expect to get in the future and consider important to do well on.

Size of Dev and Test Sets

<Old way of Splitting data>



<Size of Test Set>

→ Set your test set to be big enough to give high confidence in the overall performance of your system.

to your purpose..

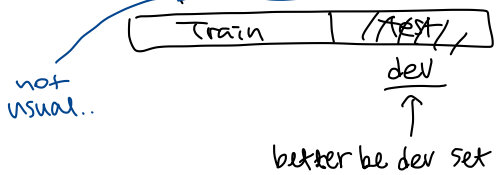
10,000

100,000 ...

.

Some applications: don't need high confidence

: Train + Dev. No test set (ok)



When to Change dev/test sets and metrics

<Cat dataset Examples>

time to think about defining new evaluation metric

→ Metric: classification error

Algorithm A: 3% error → pornographic images (totally unacceptable)

✓ Algorithm B: 5% error

B is better ∵ not learning though any pornographic images.

(even A was better evaluation metric)

↳ no longer correctly rank ordering performance. miss predicting

Metric + dev: prefer A (lower error)

You/Users : prefer B

(just one possible way)

$$\text{Error}: \frac{1}{\sum w^{(i)}} \sum_{i=1}^{m_{dev}} w^{(i)} I\{y_{pred}^{(i)} \neq y^{(i)}\}$$

this formula just count up the number of misclassifying examples.

predicted value. (0/1)

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

give a much large weight to examples that are porn.

<Orthogonalization for cat pictures: anti-porn>

→ 1. So far we've only discussed how to define a metric to evaluate classifiers. place the target. ↕ distinct step

→ 2. Worry separately about how to do well on this metric. aim/shoot at the target.

$$\rightarrow J = \frac{1}{\sum w^{(i)}} \sum_{i=1}^{m} w^{(i)} \mathcal{L}(\hat{y}^{(i)}, y^{(i)})$$

changing cost function J that your network is optimizing.

✓ If doing well on your metric + dev/test set does not correspond to doing well on your application, change your metric and/or dev/test set.