

# Copyright Notice

These slides are distributed under the Creative Commons License.

[DeepLearning.AI](#) makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite [DeepLearning.AI](#) as the source of the slides.

For the rest of the details of the license, see <https://creativecommons.org/licenses/by-sa/2.0/legalcode>



deeplearning.ai

# Introduction to ML strategy

---

## Why ML Strategy?

# Motivating example



90%

## Ideas:

- Collect more data ←
- Collect more diverse training set
- Train algorithm longer with gradient descent
- Try Adam instead of gradient descent
- Try bigger network
- Try smaller network
- Try dropout
- Add  $L_2$  regularization
- Network architecture
  - Activation functions
  - # hidden units
  - ...



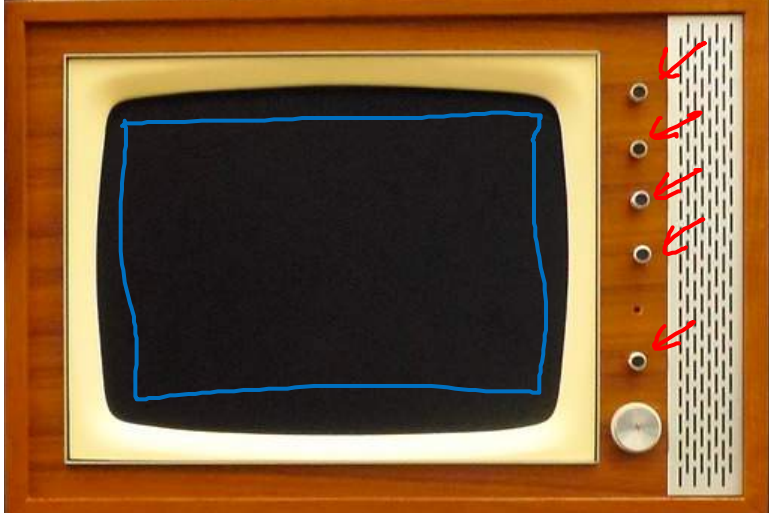
deeplearning.ai

# Introduction to ML strategy

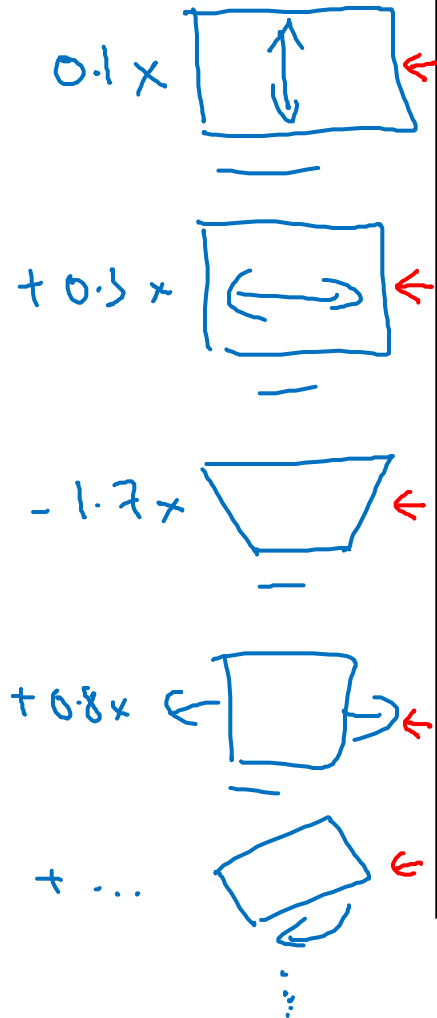
---

## Orthogonalization

# TV tuning example



Orthogonalization



## Car

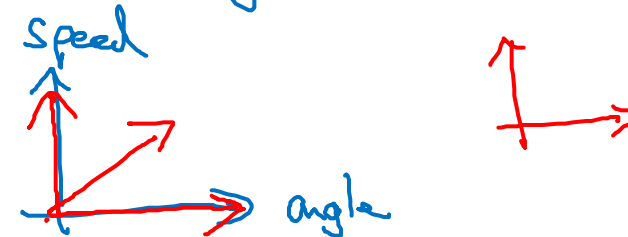


$\rightarrow$  Steering]

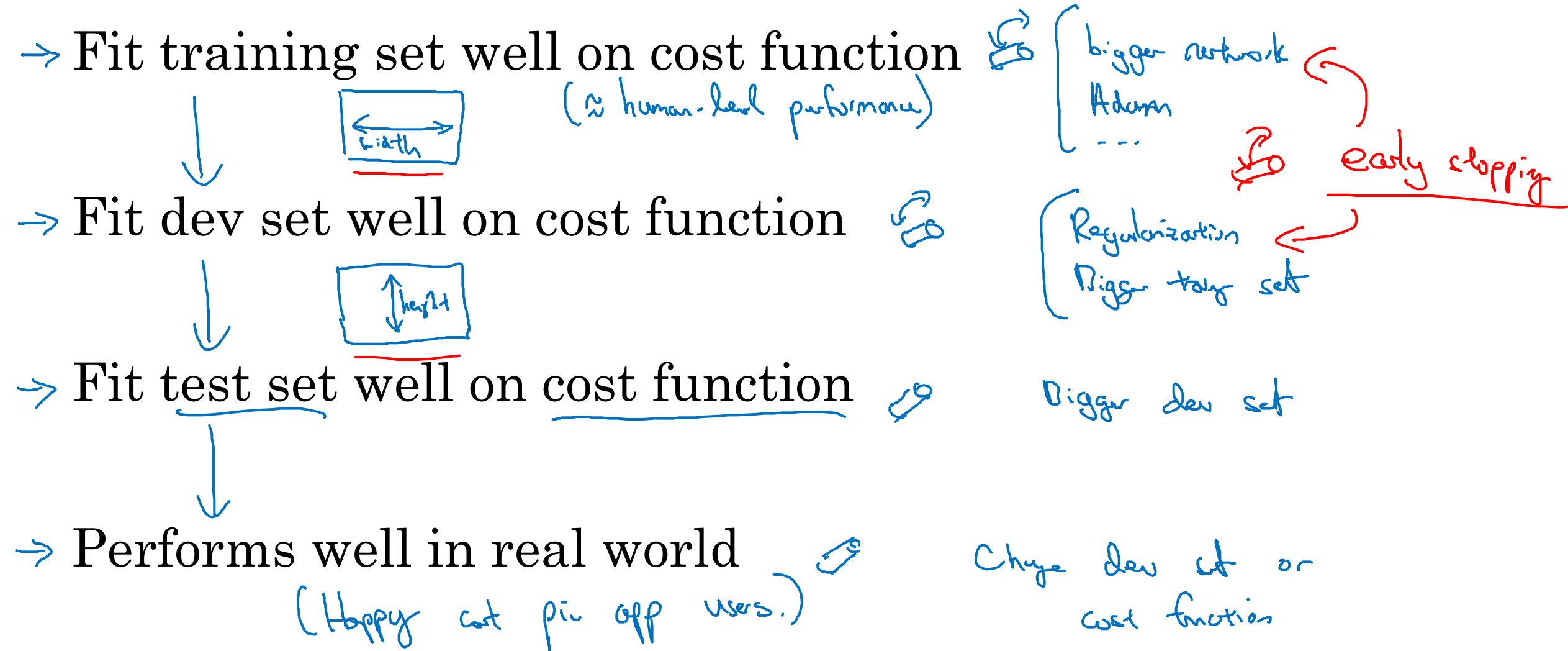
$\rightarrow$  { Accelerate  
Braking }

$\rightarrow \underline{0.3 \times \text{angle} - 0.8 \text{ speed}}$

$\rightarrow 2 \times \text{angle} + 0.9 \text{ speed}$



# Chain of assumptions in ML





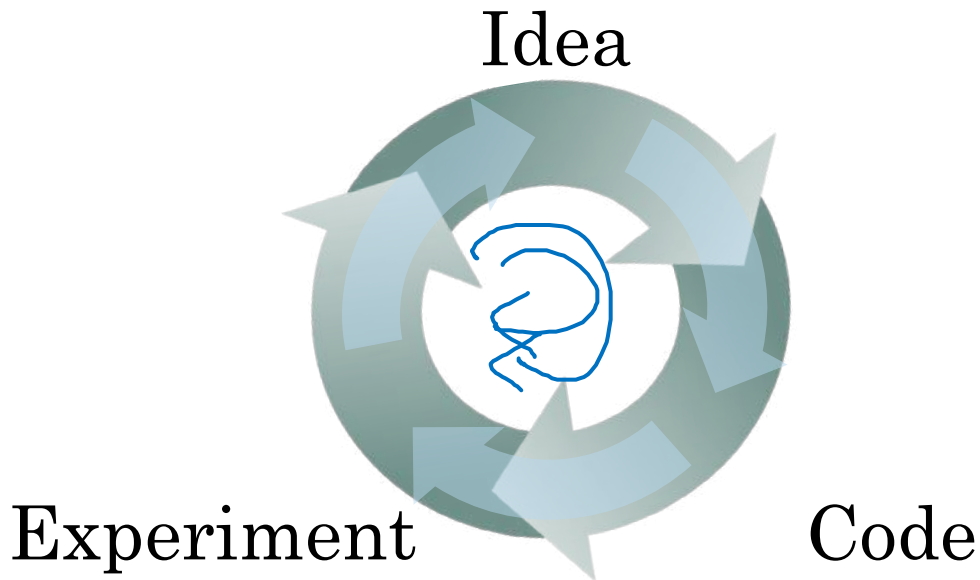
deeplearning.ai

Setting up  
your goal

---

Single number  
evaluation metric

# Using a single number evaluation metric



→ Of examples recognized as cost,  
what % actually are costs?

→ what % of actual costs  
are correctly recognized

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

F<sub>1</sub> score = "Average" of P and R.

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

Dev set + Single number evaluation metric  
real speed up iterating



# Another example

Algorithm	US	China	India	Other
A	<u>3%</u>	7%	5%	9%
B	5%	6%	5%	10%
C	2%	3%	4%	5%
D	5%	8%	7%	2%
E	4%	5%	2%	4%
F	7%	11%	8%	12%





deeplearning.ai

Setting up  
your goal

---

Satisficing and  
optimizing metrics

# Another cat classification example

Classifier	Accuracy	Running time
A	90%	80ms
B	92%	95ms
C	95%	1,500ms

$$\text{Cost} = \text{accuracy} - 0.5 \times \text{Running Time}$$

maximize accuracy

subject to Running Time  $\leq$  100 ms.

N metrics : 1 optimizing  
N-1 satisficing

Wakewords / Trigger words

Alexa, OK Google,

Hey Siri, nihao baidu  
你好百度

accuracy.

#false positive

maximize accuracy.

s.t.  $\leq$  1 false positive  
every 24 hours.



deeplearning.ai

Setting up  
your goal

---

Train/dev/test  
distributions

# Cat classification dev/test sets

development set, hold out cross validation set

Regions:

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

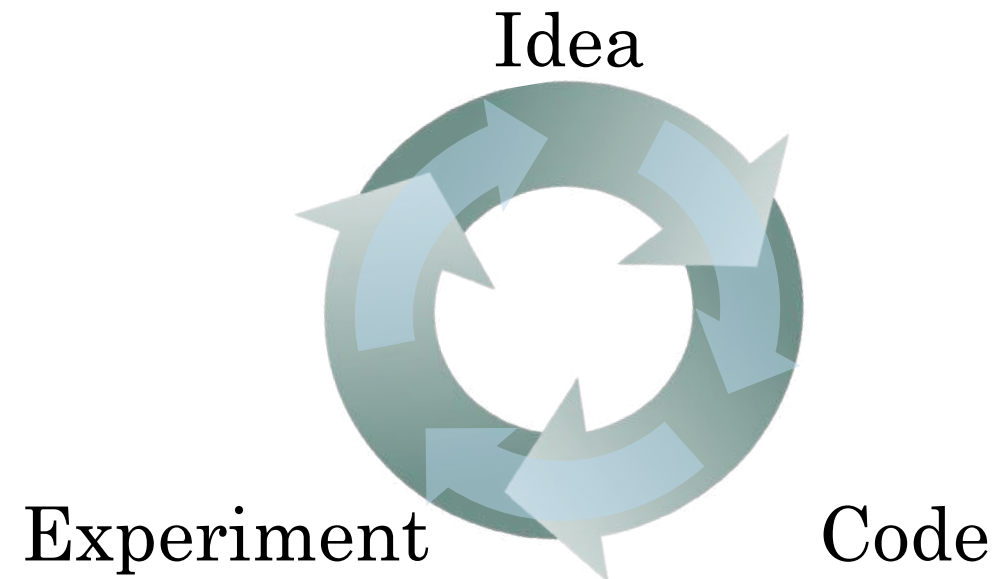
Dev

Test

→ Randomly shuffle into dev/test



dev set  
+  
metric



# True story (details changed)

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



$x \rightarrow y$  (repay loan?)



[ Tested on low income zip codes

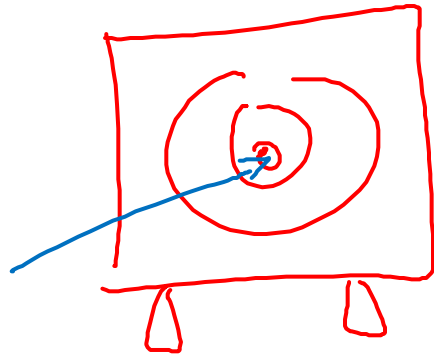
$\sim 3$  month



# Guideline

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

training



dev  
metric

test



deeplearning.ai

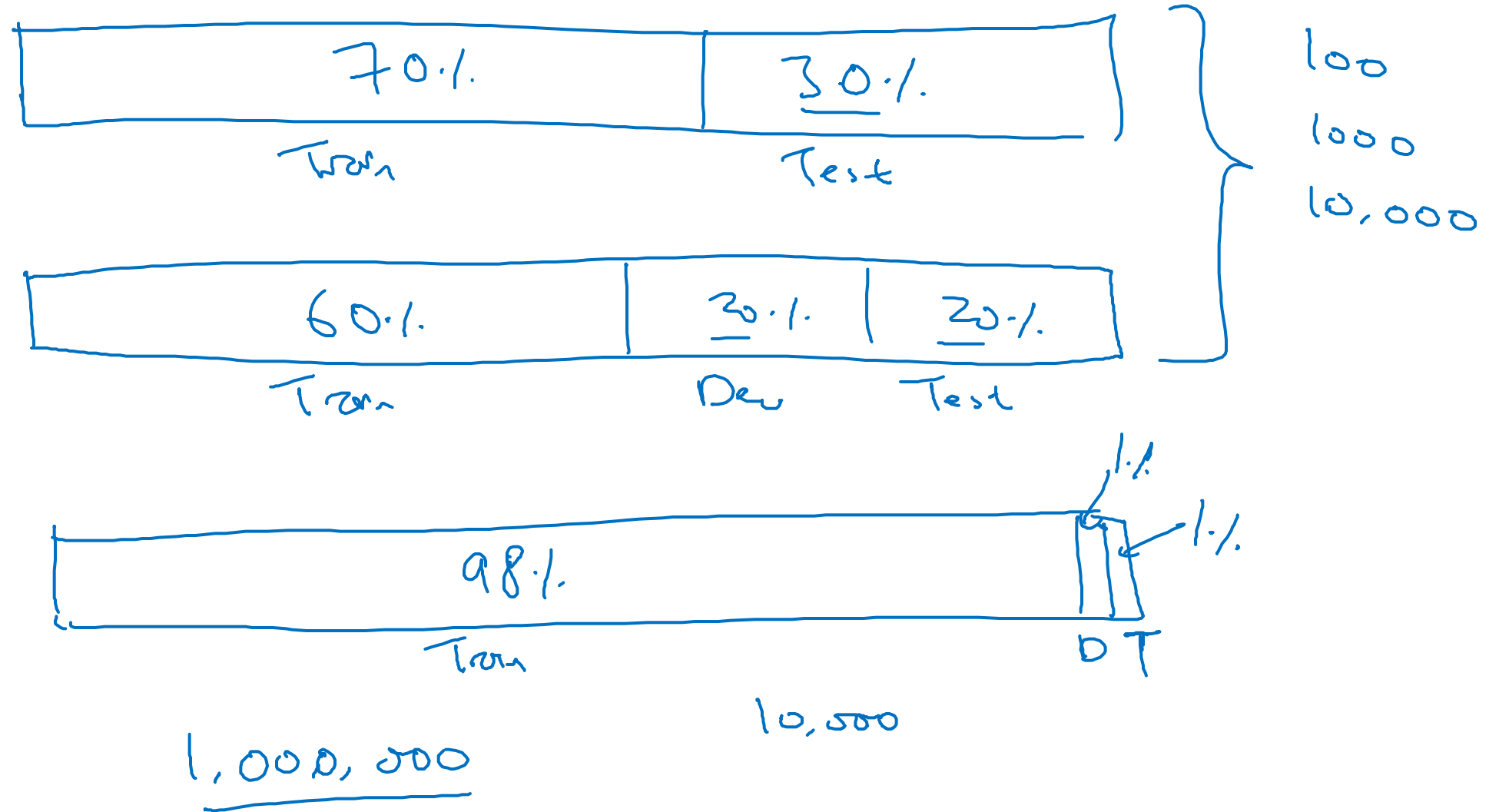
Setting up  
your goal

---

Size of dev  
and test sets



# Old way of splitting data



# Size of dev set

A B

Set your dev set to be big enough to detect differences in  
algorithm/models you're trying out.

100 : small  
└ 1%

1,000

10,000

100,000

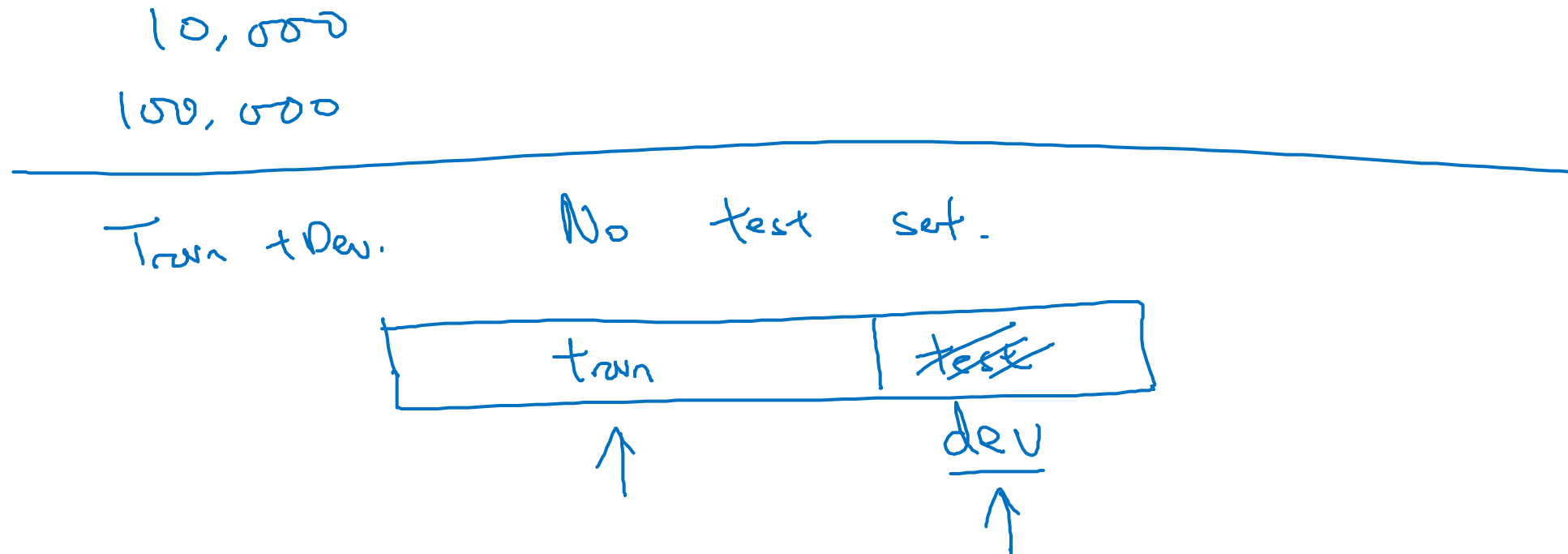
<sup>A</sup> 97% → <sup>B</sup> 97.1%  
0.1%  
└

0.01%  
└  
0.001%

Online advertising

# Size of test set

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





deeplearning.ai

Setting up  
your goal

---

When to change  
dev/test sets and  
metrics

# Cat dataset examples

Metric + Dev : Prefer A  
You/users : Prefer B.

→ Metric: classification error

Algorithm A: 3% error

→ pornographic

✓ Algorithm B: 5% error

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

↙ predicted value (0/1)

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

# Orthogonalization for cat pictures: anti-porn

- 1. So far we've only discussed how to define a metric to evaluate classifiers. ← Place target ↗
- 2. Worry separately about how to do well on this metric. ↗
- ↖ Aim (shoot at target)

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



# Another example

Algorithm A: 3% error

✓ Algorithm B: 5% error ←

→ Dev/test



→ User images



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.



deeplearning.ai

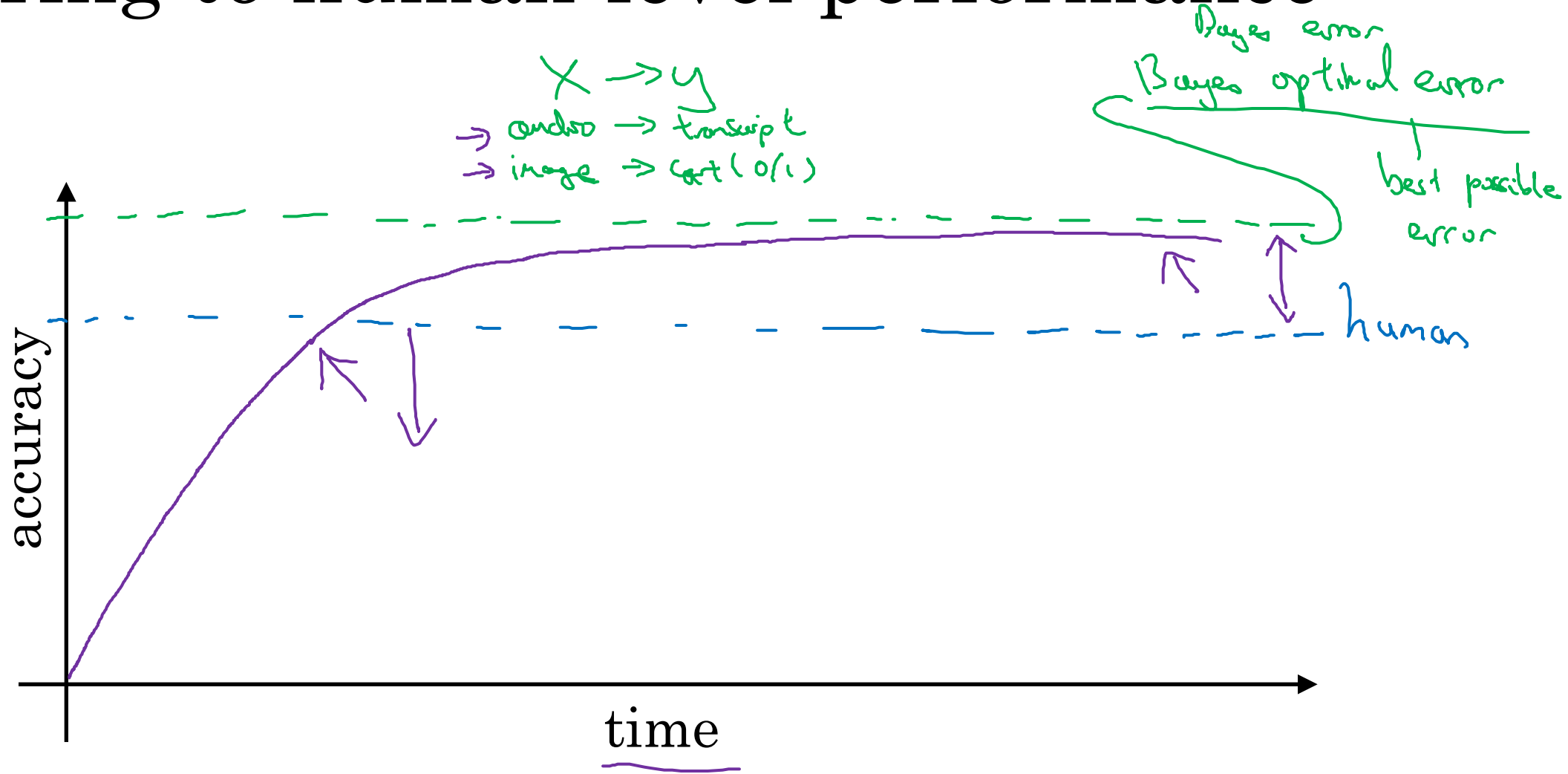
Comparing to human-level performance

---

Why human-level performance?



# Comparing to human-level performance



# Why compare to human-level performance

Humans are quite good at a lot of tasks. So long as ML is worse than humans, you can:

- - Get labeled data from humans.  $(x, y)$
- - Gain insight from manual error analysis:  
Why did a person get this right?
- - Better analysis of bias/variance.



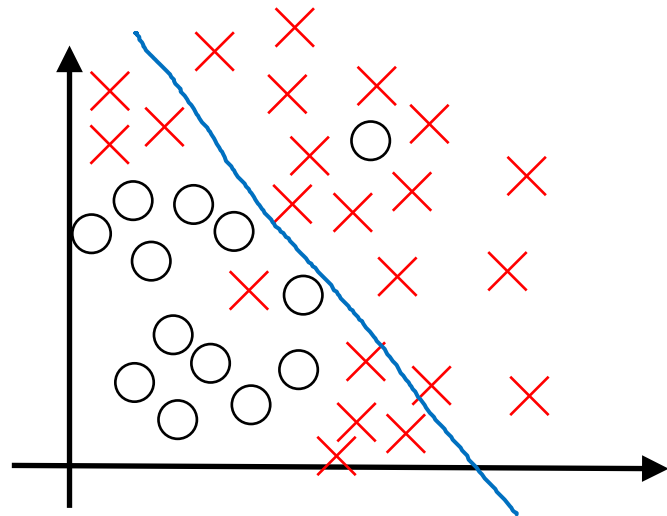
deeplearning.ai

Comparing to human-  
level performance

---

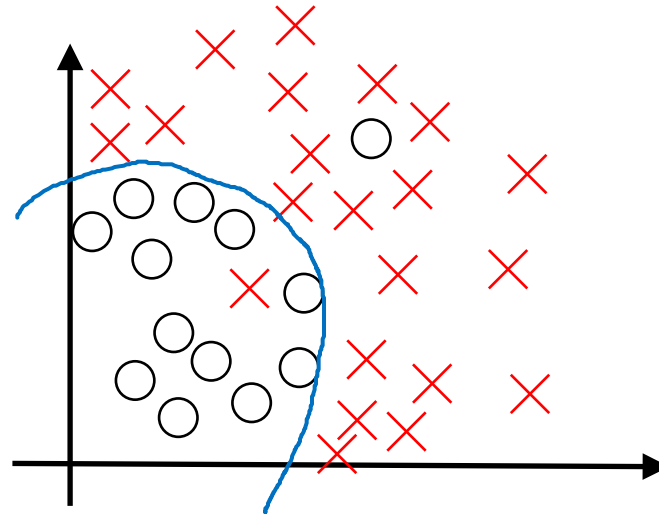
**Avoidable bias**

# Bias and Variance

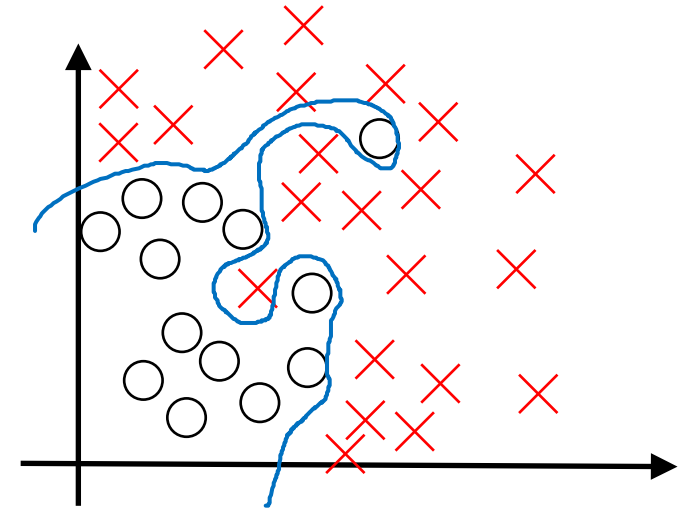


high bias

*underfitting*



"just right"



high variance

*overfitting*

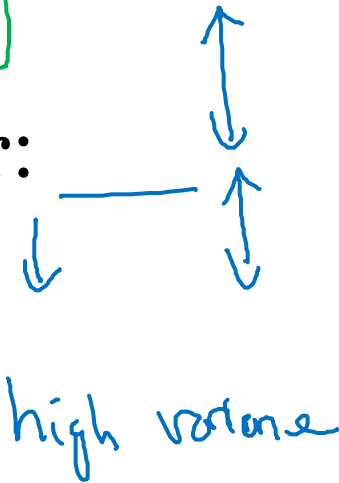
# Bias and Variance

Cat classification

Human-level  $\approx 0\%$  ----

Training set error:

Dev set error:



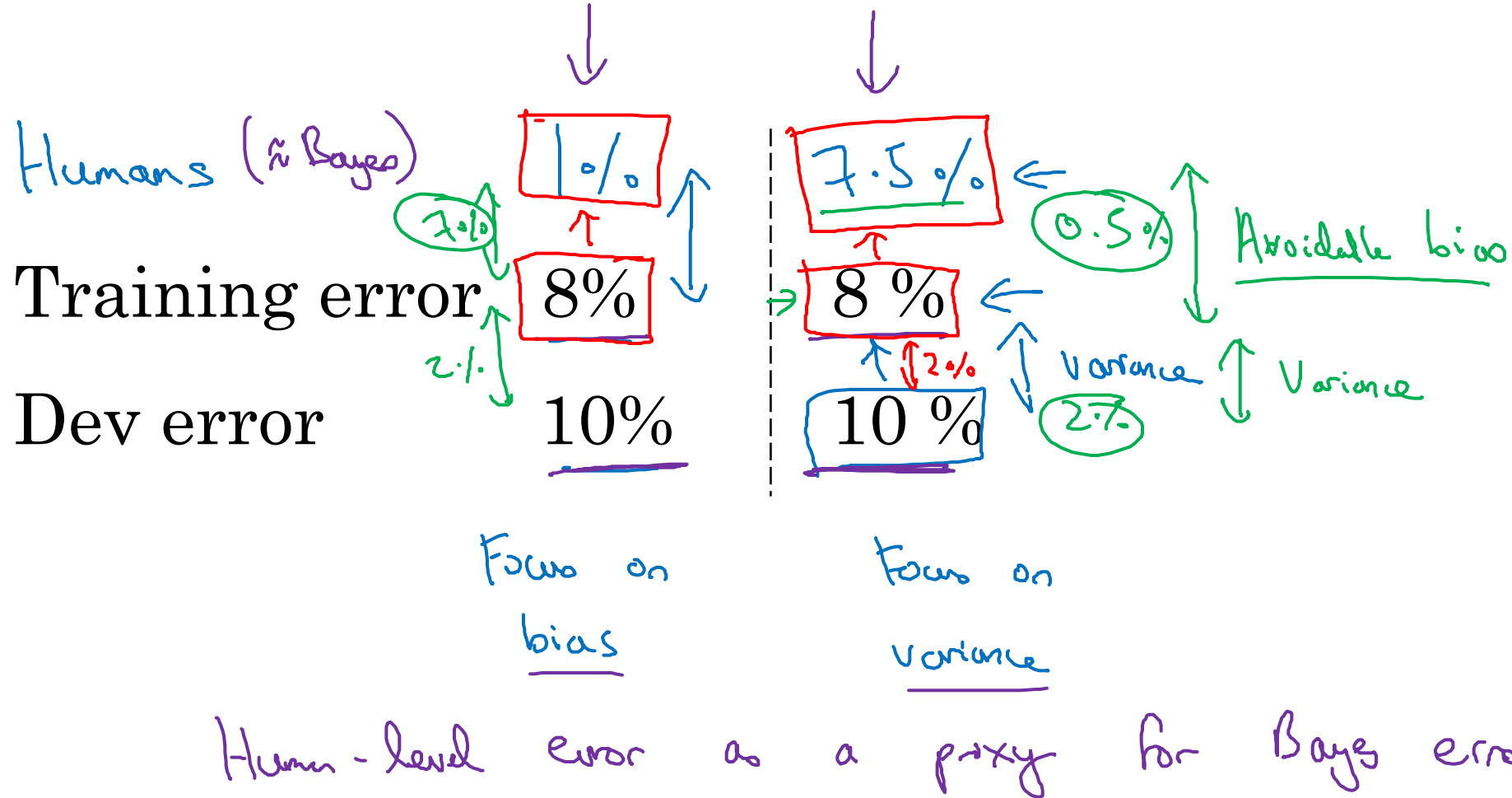
high bias



high bias  
high variance

low bias  
low variance

# Cat classification example





deeplearning.ai

Comparing to human-  
level performance

---

Understanding  
human-level  
performance

# Human-level error as a proxy for Bayes error

Medical image classification example:

Suppose:

(a) Typical human ..... 3 % error

→ (b) Typical doctor ..... 1 % error

(c) Experienced doctor ..... 0.7 % error

→ (d) Team of experienced doctors .. 0.5 % error ←

Bayes error  $\leq$  0.5 %

What is “human-level” error?





# Error analysis example

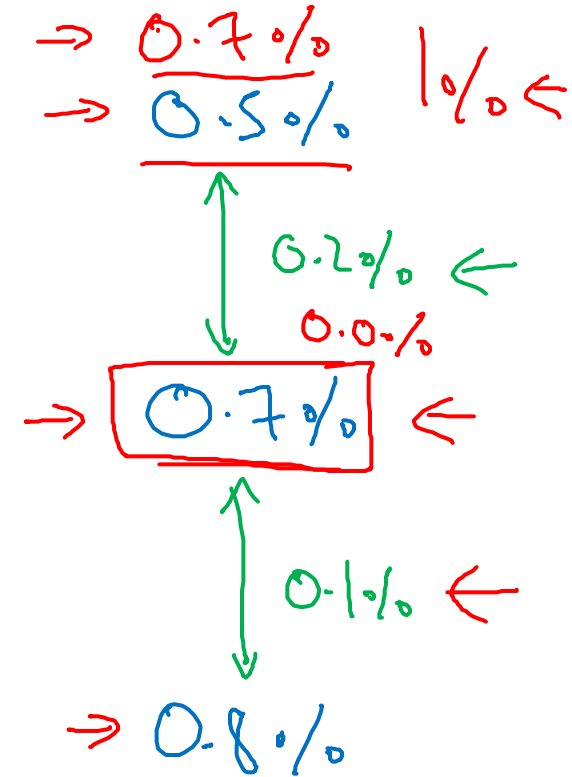
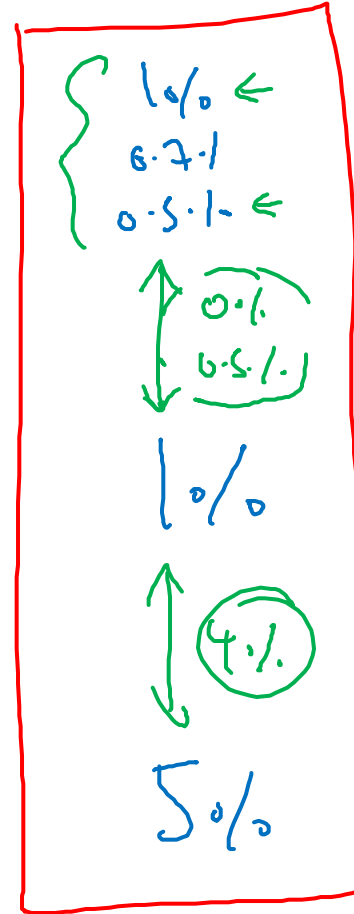
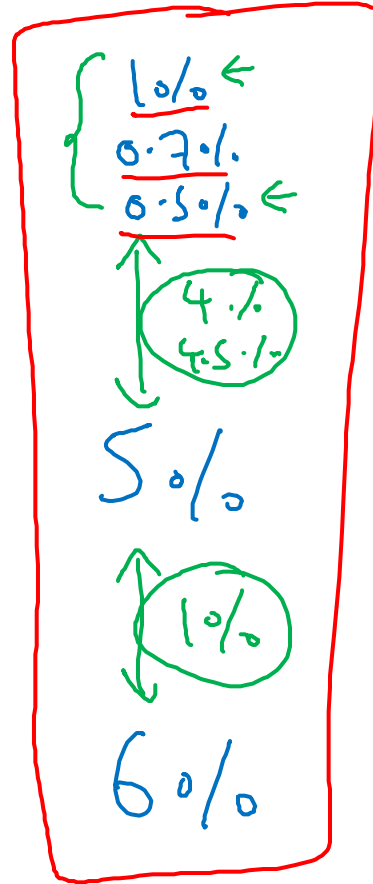
Human (proxy for Bayes error)

↑ Avoidable bias  
↓

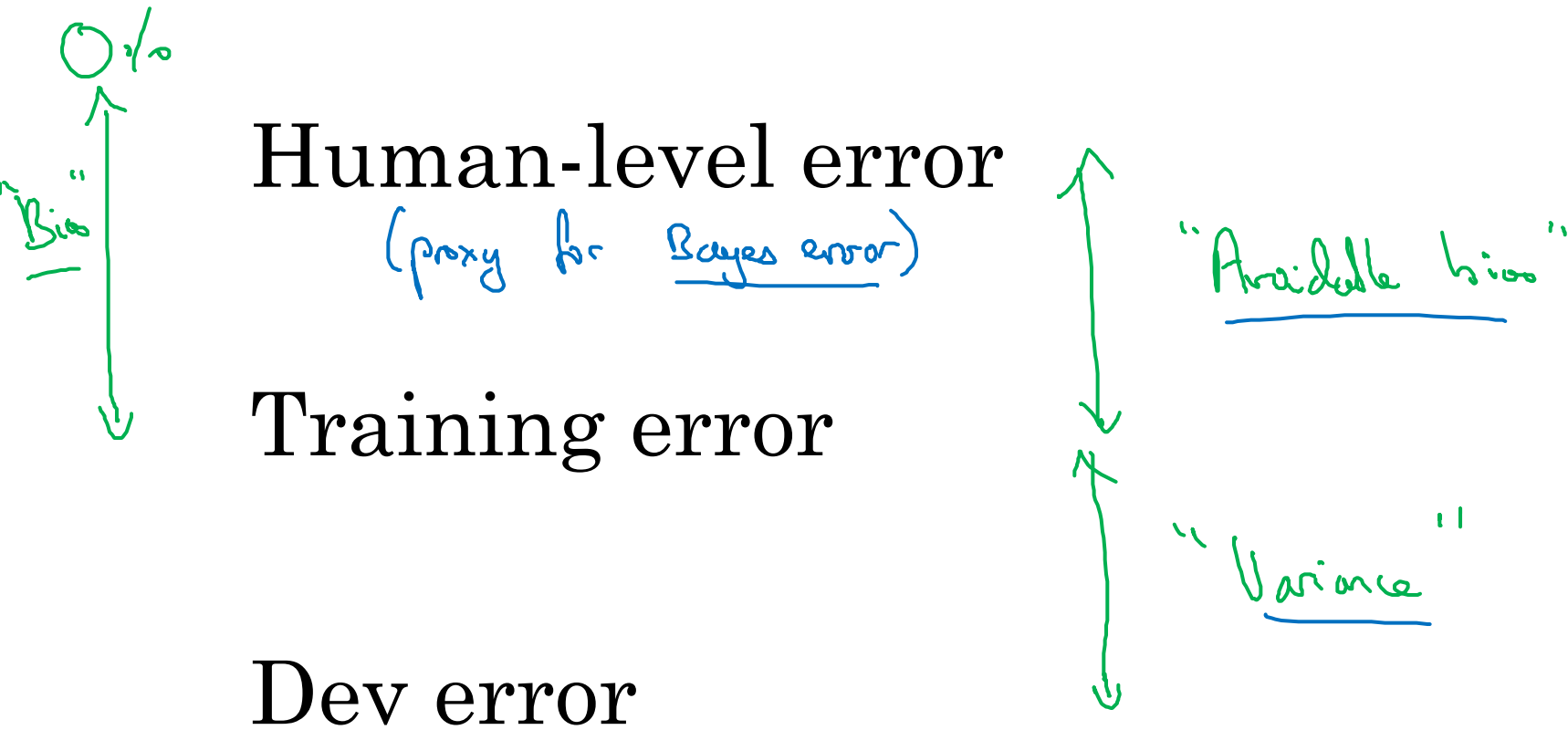
Training error

↑ Variance  
↓

Dev error



# Summary of bias/variance with human-level performance





deeplearning.ai

Comparing to human-  
level performance

---

Surpassing human-  
level performance

# Surpassing human-level performance

Team of humans

0.5%

One human

0.1  $\updownarrow$  ~~1.0%~~

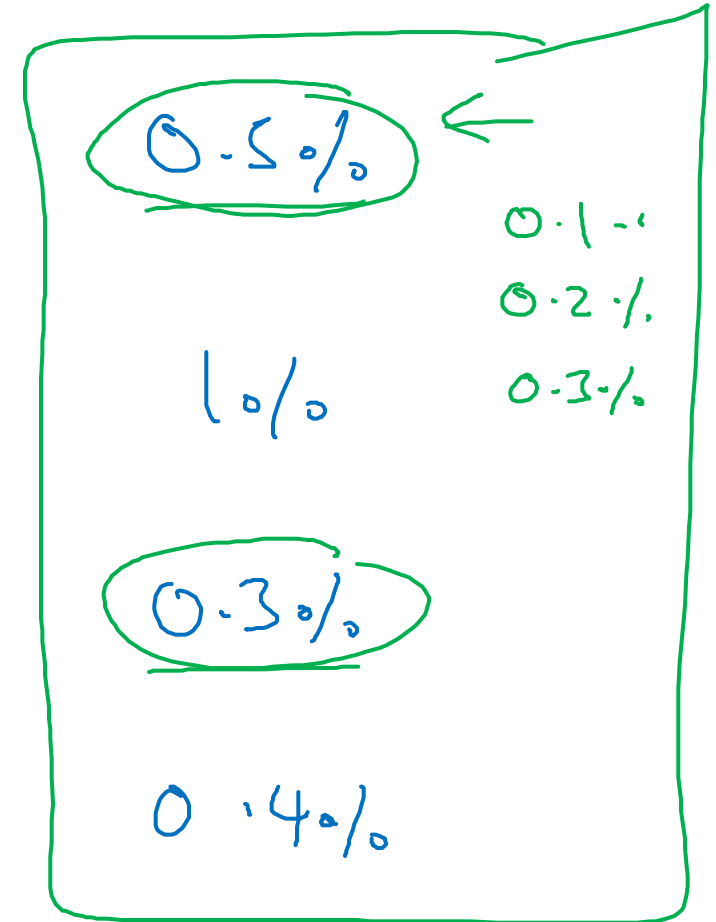
Training error

0.6%

Dev error

0.2  $\updownarrow$  0.8%

What is avoidable bias?



# Problems where ML significantly surpasses human-level performance

- - Online advertising
- - Product recommendations
- - Logistics (predicting transit time)
- - Loan approvals

Structured data

Not natural perception

Lots of data

- Speech recognition
- Some image recognition
- Medical
  - ECG, Skin cancer, ...

Humans are better at 'natural perception' data not so much at other structured, statistical types of data.



deeplearning.ai

Comparing to human-  
level performance

---

Improving your model  
performance

# The two fundamental assumptions of supervised learning

1. You can fit the training set pretty well.



~ Avoidable bias

2. The training set performance generalizes pretty well to the dev/test set.



~ Variance

# Reducing (avoidable) bias and variance

Human-level



Avoidable bias

Training error



Variance

Dev error

Train bigger model

Train longer/better optimization algorithms

- momentum, RMSprop, Adam

NN architecture/hyperparameters search

RNN  
CNN

More data

Regularization

-  $L_2$ , dropout, data augmentation

NN architecture/hyperparameters search



# Copyright Notice

These slides are distributed under the Creative Commons License.

[DeepLearning.AI](#) makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite [DeepLearning.AI](#) as the source of the slides.

For the rest of the details of the license, see <https://creativecommons.org/licenses/by-sa/2.0/legalcode>



deeplearning.ai

# Error Analysis

---

Carrying out error  
analysis

# Look at dev examples to evaluate ideas



90% accuracy  
→ 10% error

Should you try to make your cat classifier do better on dogs? ←

Error analysis:

- Get ~100 mislabeled dev set examples. → 5-10 min
- Count up how many are dogs.

→ 5%  
5/100

10%  
↓  
9.5%

"ceiling"

→ 50%  
50/100

10%  
↓  
5%

# Evaluate multiple ideas in parallel

Ideas for cat detection:

- Fix pictures of dogs being recognized as cats ←
- Fix great cats (lions, panthers, etc..) being misrecognized ←
- Improve performance on blurry images ←

Image	Dog	Great Cats	Blurry	Instagram	Comments
1	✓			✓	Pitbull
2			✓	✓	
3		✓	✓		Rainy day at zoo
⋮	⋮	⋮	⋮	⋮	
% of total	<u>8%</u>	<u>43%</u>	<u>61%</u>	<u>12%</u>	










deeplearning.ai

# Error Analysis

---

## Cleaning up Incorrectly labeled data

# Incorrectly labeled examples

x							
y	<u>1</u>	<u>0</u>	<u>1</u>	<u>1</u>	<u>0</u>	<u>1</u>	1

Training Set.

The 7th example (white puppy) is highlighted with a blue box and an arrow pointing to its label '1', indicating it is an incorrectly labeled example.

DL algorithms are quite robust to random errors in the training set.

Systematic errors

# Error analysis

✓

Image	Dog	Great Cat	Blurry	Incorrectly labeled	Comments
...					
98				✓	Labeler missed cat in background
99		✓			
100				✓	Drawing of a cat; Not a real cat.
% of total	<u>8%</u>	<u>43%</u>	<u>61%</u>	<u>6%</u>	

↑  
↓

←

←

Overall dev set error ..... 10%

Errors due incorrect labels ..... 0.6% ←

Errors due to other causes ..... 9.4% ←

↑

2.1%

1.9%

✓  
2.0%  
✓  
0.6%  
1.4%  
2.1%

Goal of dev set is to help you select between two classifiers A & B.

# Correcting incorrect dev/test set examples

- Apply same process to your dev and test sets to make sure they continue to come from the same distribution
- Consider examining examples your algorithm got right as well as ones it got wrong. 20%
- Train and dev/test data may now come from slightly different distributions.





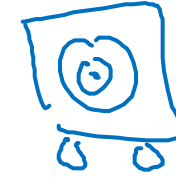
deeplearning.ai

# Error Analysis

---

Build your first system  
quickly, then iterate

# Speech recognition example



- • Noisy background
  - • Café noise
  - • Car noise

- • Accent
- • Far from mic
- • Young children's speech
- • Stuttering
- • ...

Guideline:

**Build your first  
system quickly,  
then iterate**

- • Set up dev/test set and metric
- Build initial system quickly
- Use Bias/Variance analysis & Error analysis to prioritize next steps.

- Far from mic
- Young children's speech
- Stuttering



deeplearning.ai

Mismatched training  
and dev/test data

---

Training and testing  
on different  
distributions

# Cat app example

Data from webpages



core about this  
Data from mobile app

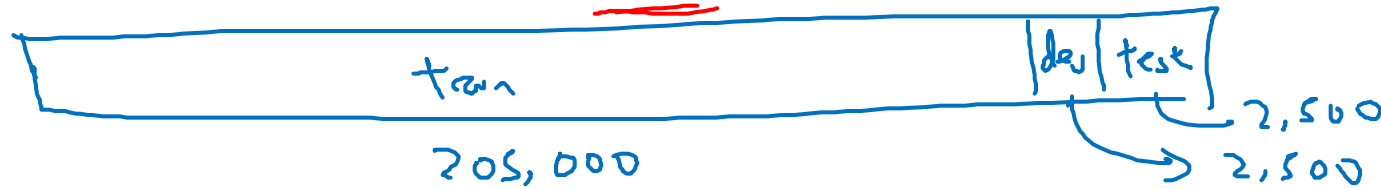


→ ≈ 200,000

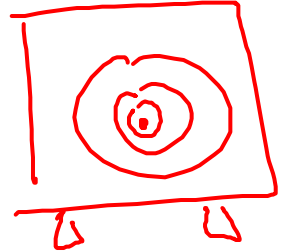
→ 210,000  
↓ shuffle

→ ≈ 10,000

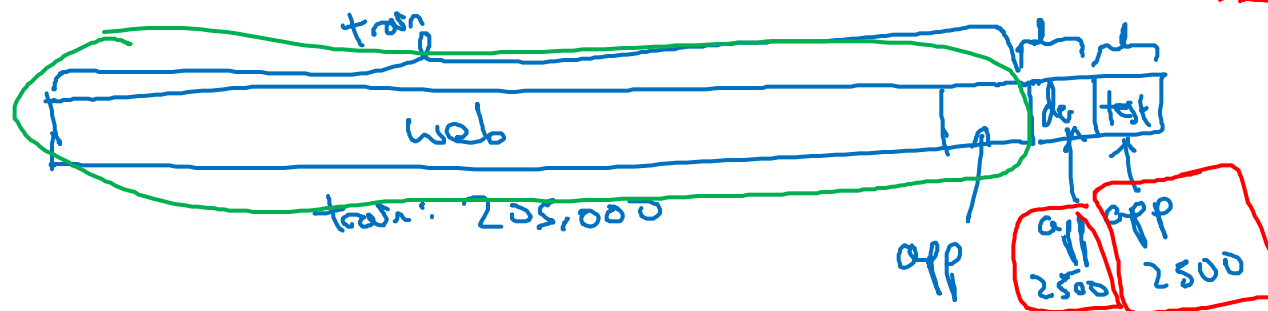
~~Option 1:~~



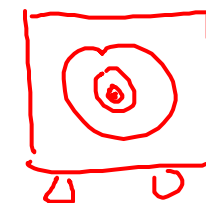
$\frac{200K}{210K}$



Option 2:



2381 - web  
119 - mobile app



# Speech recognition example

Speech activated rearview mirror



## Training

Purchased data

↓ ↓  
X, y

Smart speaker control

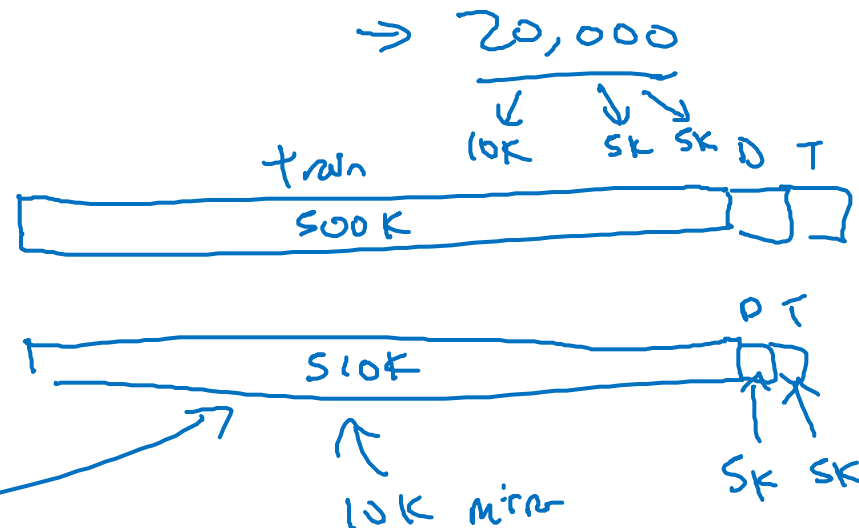
Voice keyboard

...

500,000 utterances

## Dev/test

Speech activated  
rearview mirror





deeplearning.ai

Mismatched training  
and dev/test data

---

Bias and Variance with  
mismatched data  
distributions

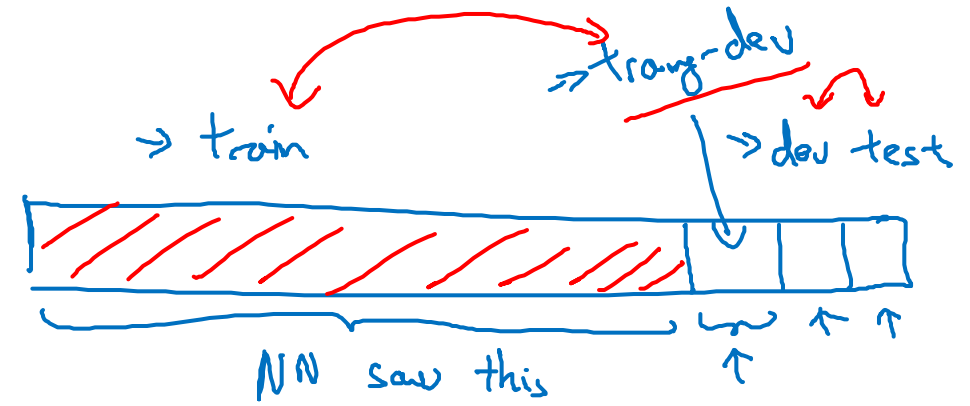
# Cat classifier example

Assume humans get  $\approx 0\%$  error.

Training error .....  $1\%$   
 Dev error .....  $10\%$   $\downarrow 9\%$

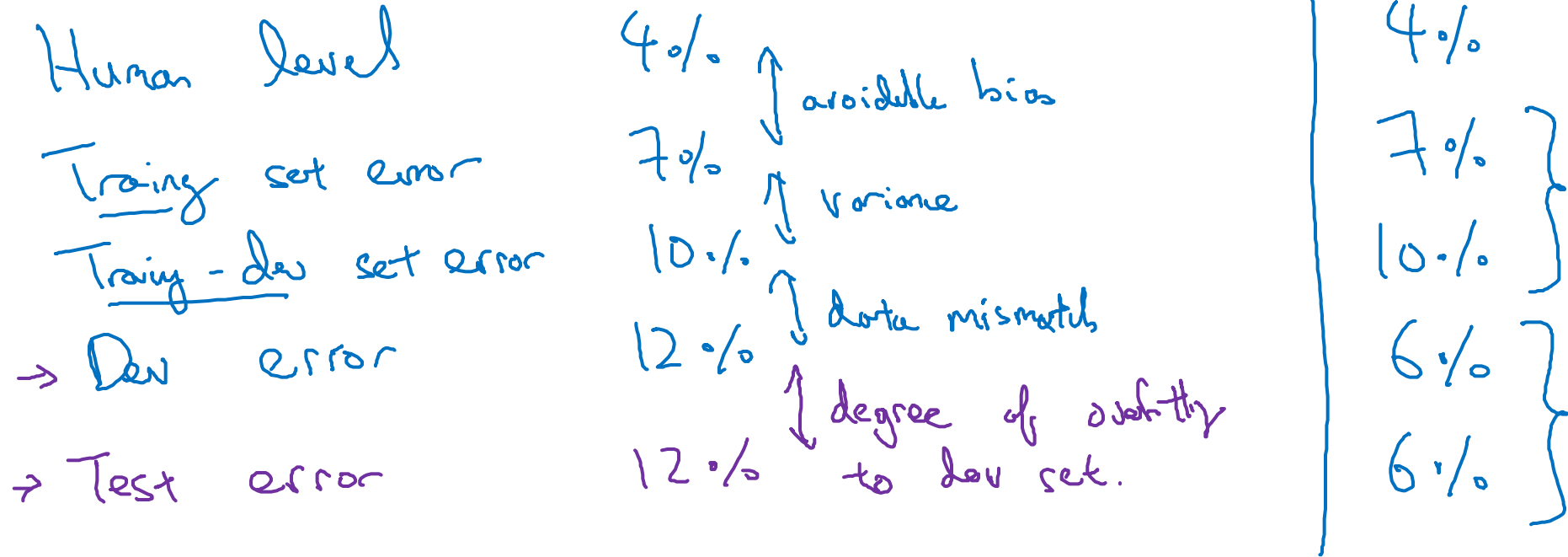
Similar to attribution - create a dataset from Training that you don't train on. This becomes pseudo 'in-between dev' error, which acts as a bridge between training error and dev error

Training-dev set: Same distribution as training set, but not used for training



Training error	$1\%$		$1\%$
→ Training-dev error	$9\%$	↑ Variance	$1.5\%$
→ Dev error	$10\%$		$10\%$
			↓ Data mismatch
		Variance	
Human error	..... $0\%$	↑ Avoidable bias	↑ Avoidable bias
Training error	$10\%$	↓ bias	$10\%$
Training-dev error	$11\%$	↑ Variance	$11\%$
Dev error	$12\%$	↑ Data mismatch	$20\%$
	Bias		Bias + Data mismatch

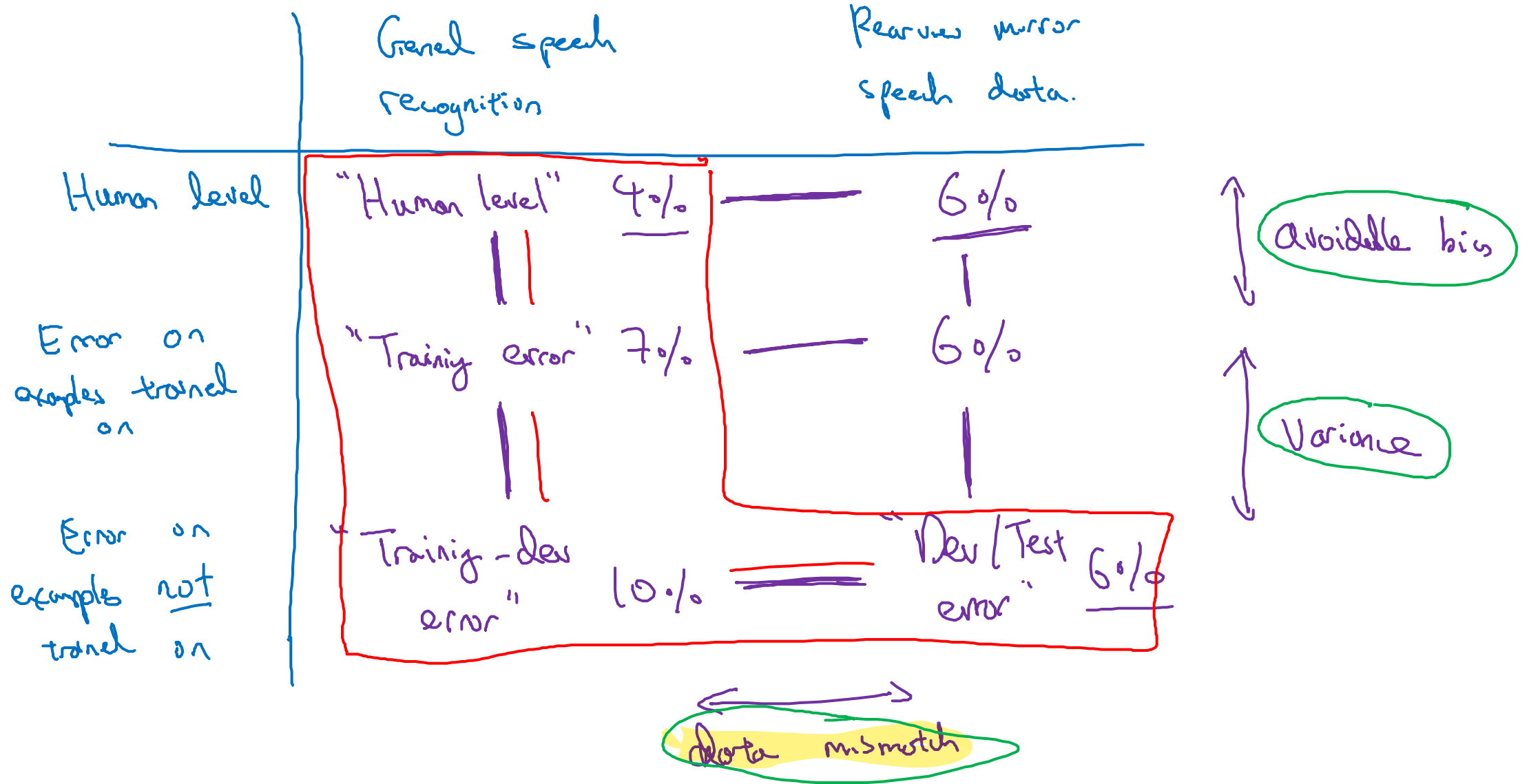
# Bias/variance on mismatched training and dev/test sets





# More general formulation

Recurrent mirror





deeplearning.ai

Mismatched training  
and dev/test data

---

Addressing data  
mismatch

# Addressing data mismatch

- • Carry out manual error analysis to try to understand difference between training and dev/test sets

E.g. noisy - car noise

street numbers

Add training data comprising, audio clips of people

- • Make training data more similar; or collect more data similar to dev/test sets

E.g. Simulate noisy in-car data

# Artificial data synthesis

Model may overfit to the specific example of car-noise.  
Getting 10,000 hours of unique car noise would be better.



+



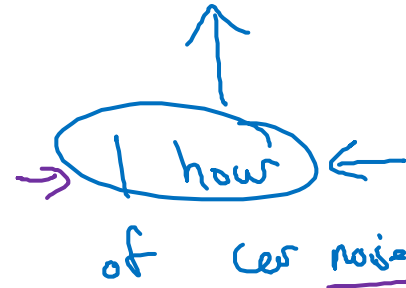
=



“The quick brown  
fox jumps  
over the lazy dog.”

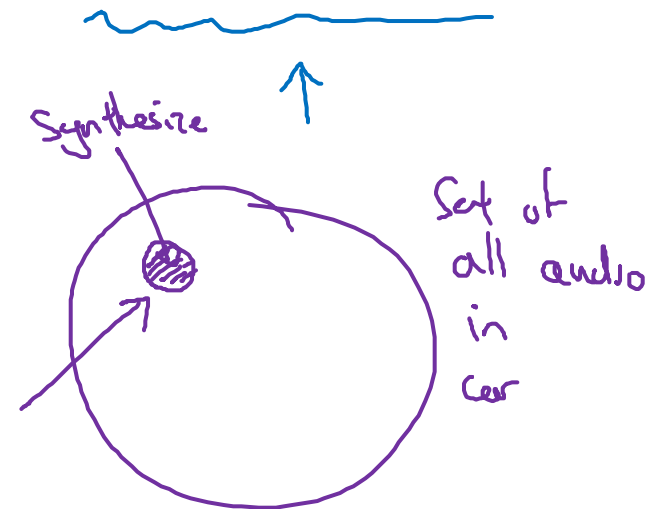
↑  
10,000 hours

Car noise



Overfit to 1 hour of  
car noise  
→ 10,000 hours ←

Synthesized  
in-car audio



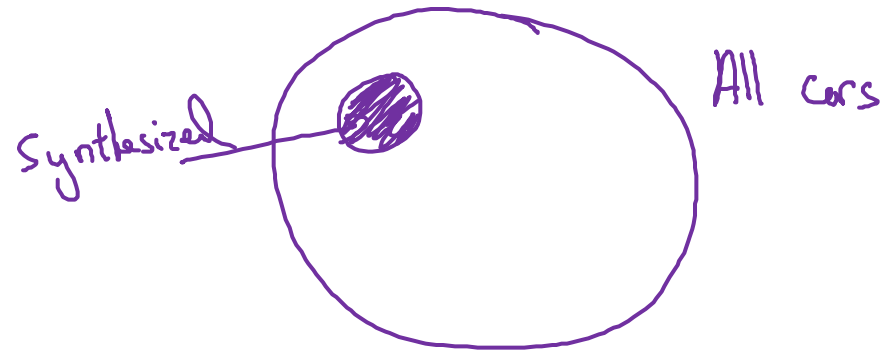
# Artificial data synthesis

Car recognition:



$\approx 20$  cars

For example, if you use data from a car simulation game, which has 20 car models - that model may not generalize well in the real world setting.





deeplearning.ai

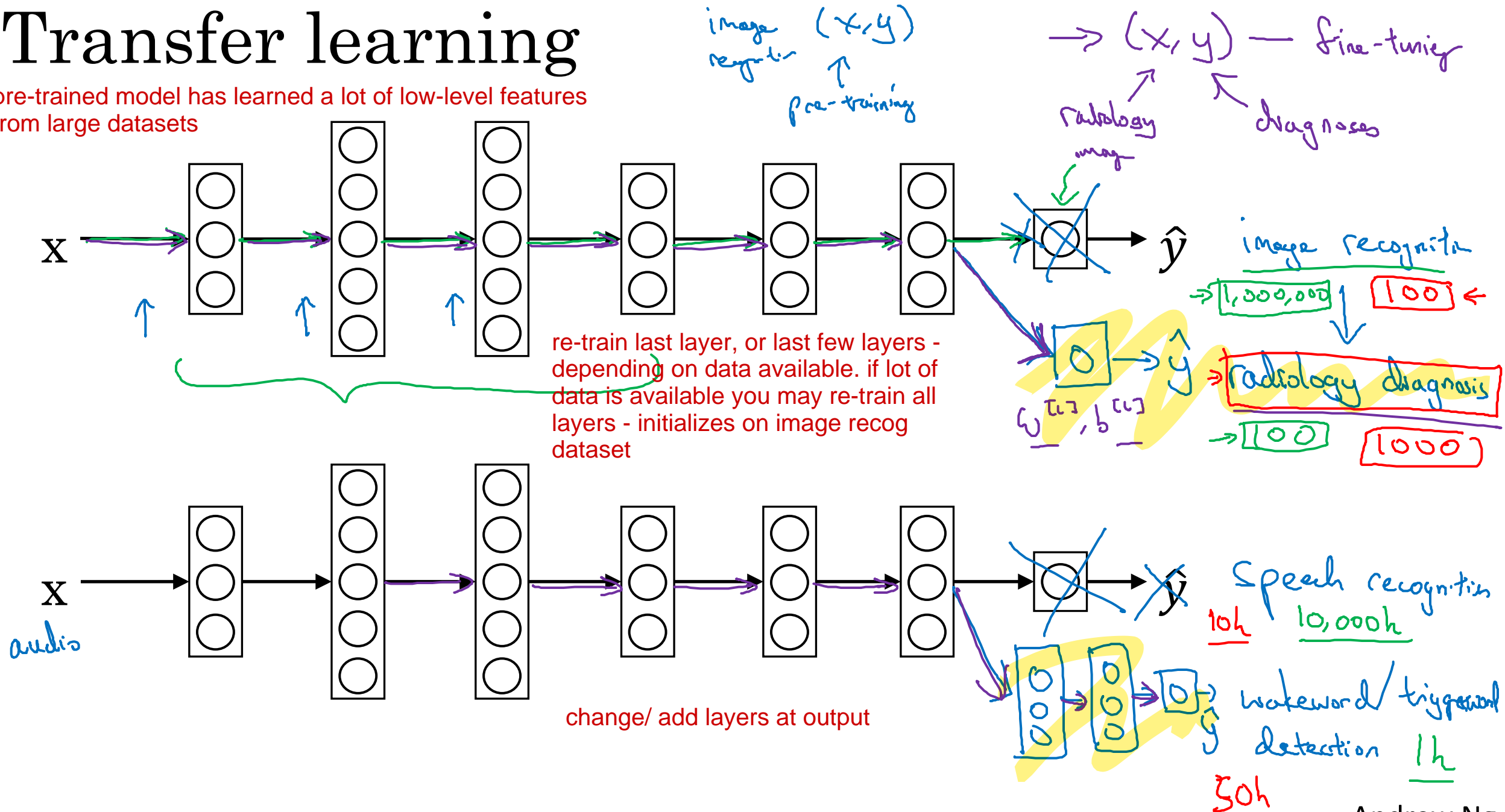
Learning from  
multiple tasks

---

Transfer learning


# Transfer learning

pre-trained model has learned a lot of low-level features from large datasets



# When transfer learning makes sense

Transfer from A  $\rightarrow$  B

- Task A and B have the same input  $x$ .
- You have a lot more data for Task A than Task B.  

- Low level features from A could be helpful for learning B.





deeplearning.ai

Learning from  
multiple tasks

---

Multi-task  
learning

# Simplified autonomous driving example



$x^{(i)}$

Pedestrians

Cars

Stop signs

Traffic lights

⋮

$y^{(i)}$

0

1

1

0

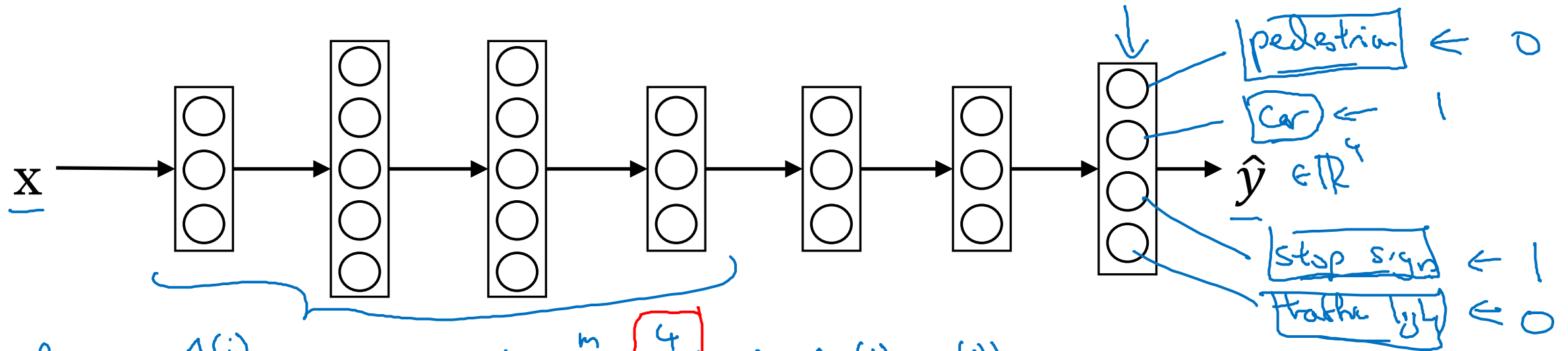
⋮

$(4, 1)$

$$Y = \begin{bmatrix} y^{(1)} & y^{(2)} & y^{(3)} & \dots & y^{(m)} \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}$$

$(4, m)$

# Neural network architecture



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

Sum only over  
value of  $j$  with  
0/1 label.

Unlike softmax regression:  
One image can have multiple labels

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

Multi-task learning  $\leftarrow$

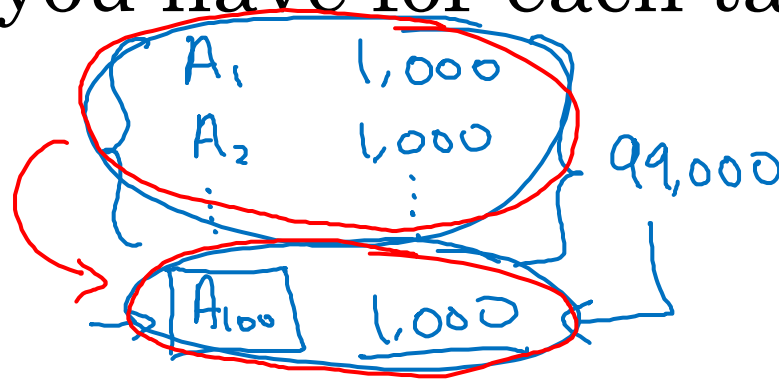
$Y = \begin{bmatrix} 1 & 1 & \dots & 1 & ? \\ 0 & 1 & \dots & 1 & ? \\ ? & ? & \dots & ? & ? \\ ? & ? & \dots & ? & ? \end{bmatrix}$

Can also use partially  
labeled data, by  
appropriate adjustment  
to loss function  
summation

# When multi-task learning makes sense

- Training on a set of tasks that could benefit from having shared lower-level features.
- Usually: Amount of data you have for each task is quite similar.

A     1,000,000  
↓     ↓  
B     1,000



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



deeplearning.ai

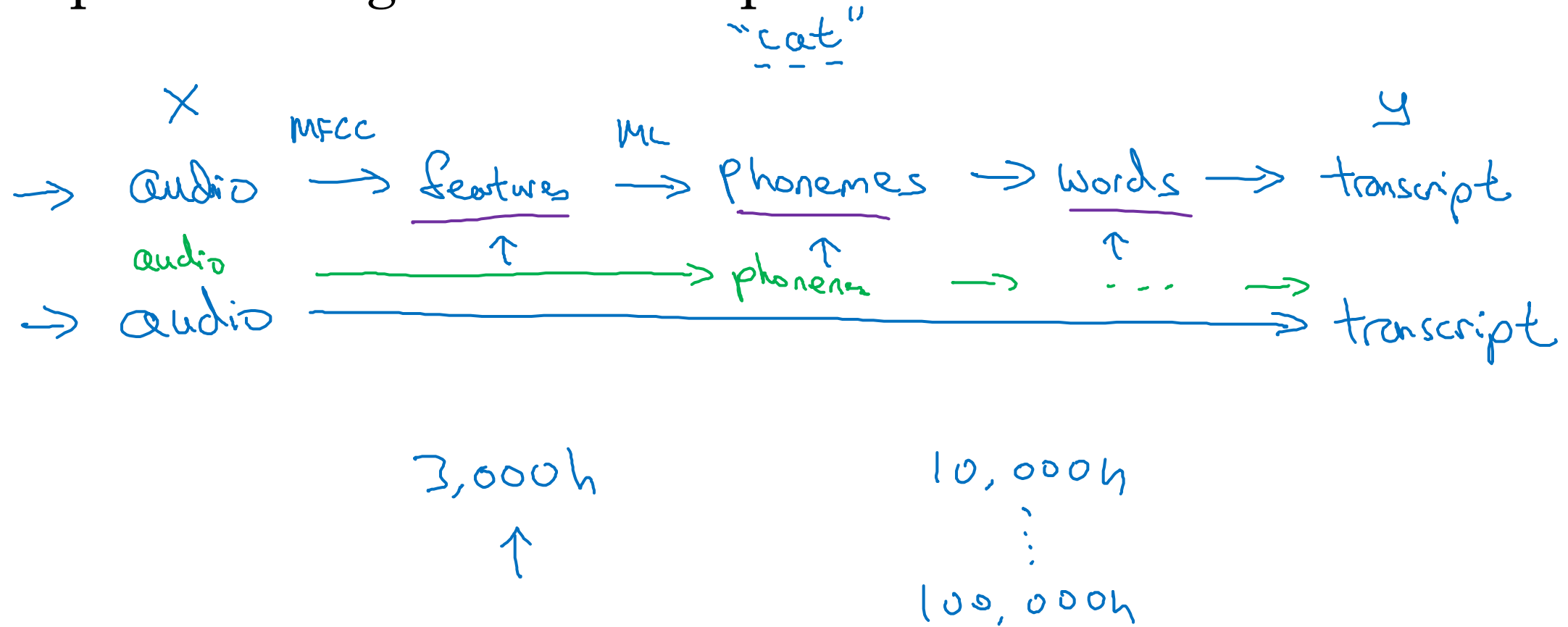
# End-to-end deep learning

---

## What is end-to-end deep learning

# What is end-to-end learning?

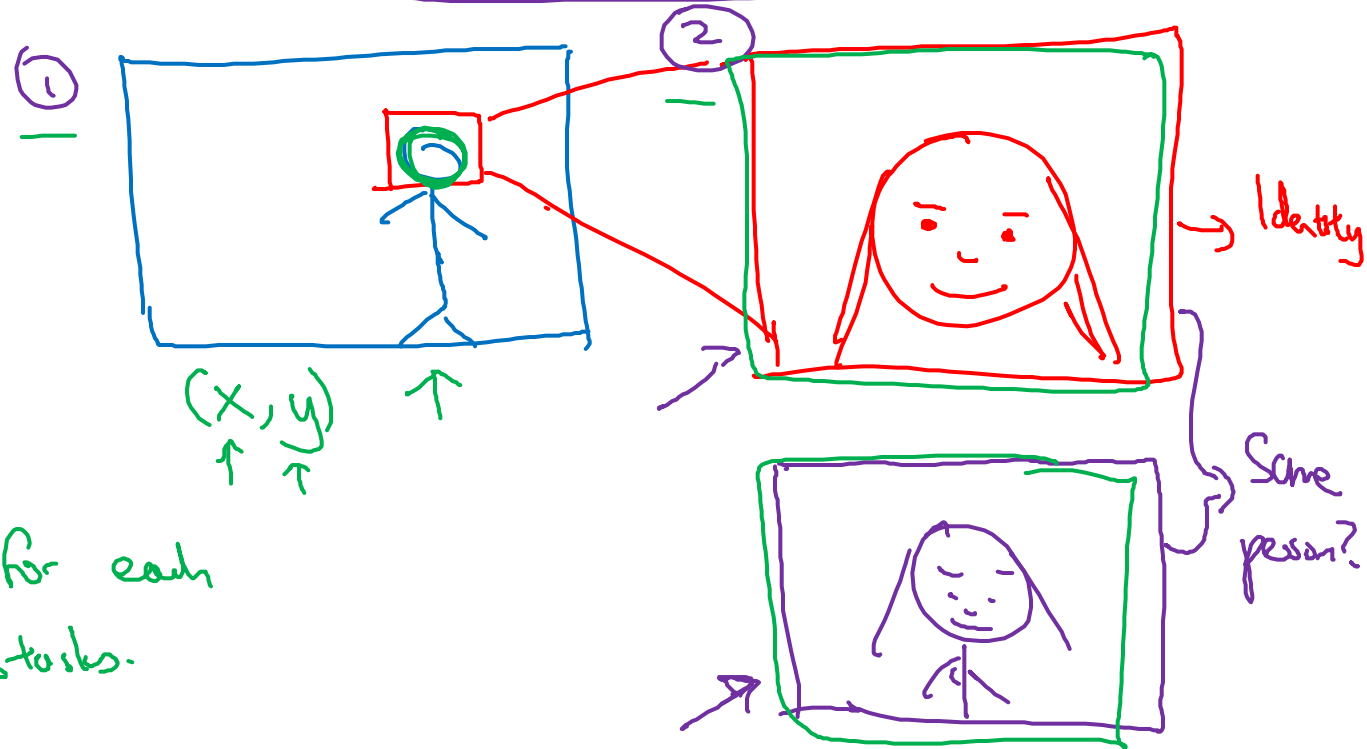
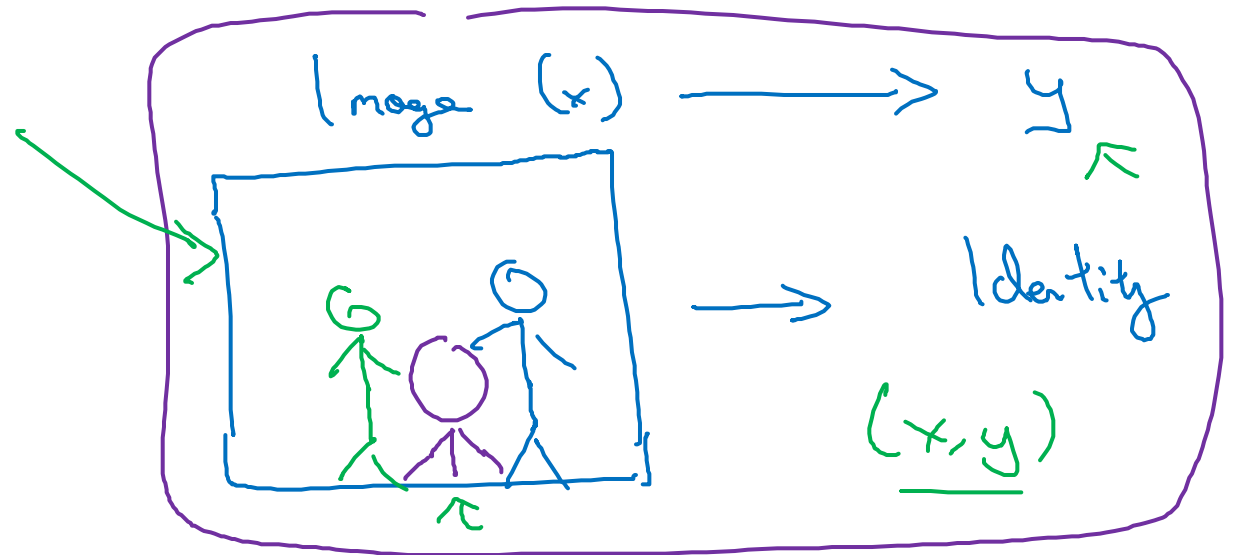
## Speech recognition example



# Face recognition



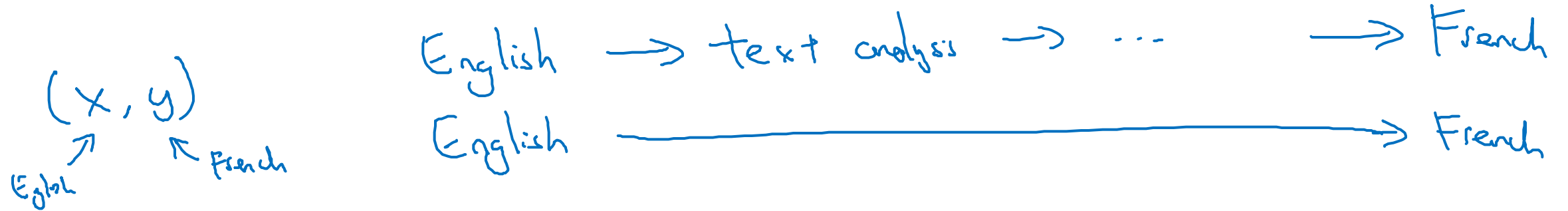
[Image courtesy of Baidu]



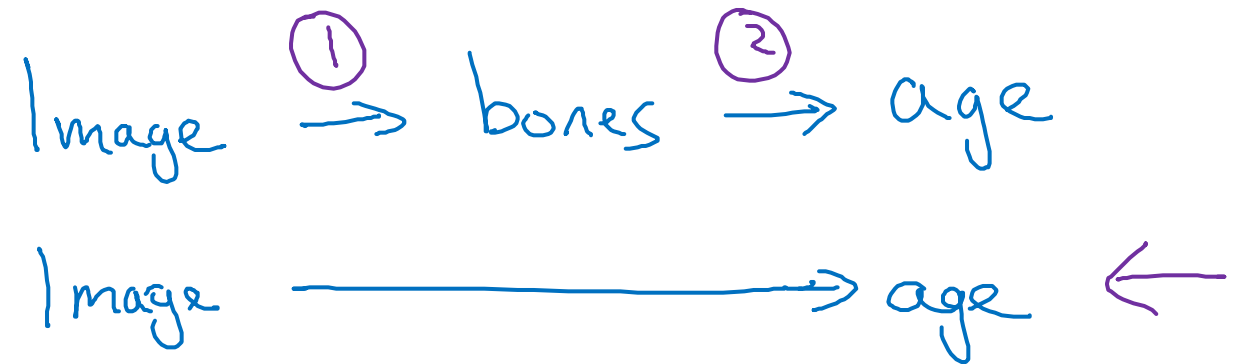
Have data for each  
of 2 sub-tasks.

# More examples

## Machine translation



## Estimating child's age:







deeplearning.ai

End-to-end deep  
learning

---

Whether to use  
end-to-end learning

# Pros and cons of end-to-end deep learning

## Pros:

- Let the data speak
- Less hand-designing of components needed

$x \rightarrow y$

→ "phonemes"  
c a t

## Cons:

- May need large amount of data
- Excludes potentially useful hand-designed components

$x - - - - - \rightarrow y$

input  
end  
↓  
 $x \rightarrow y$   
output  
end  
↓

(x, y)

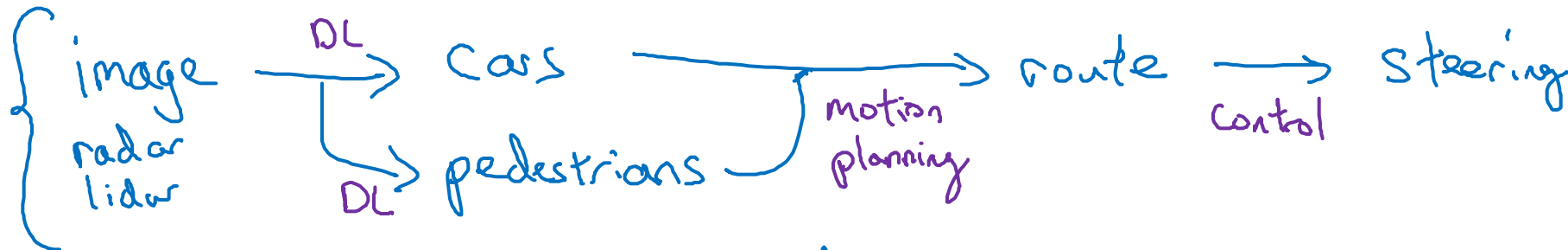
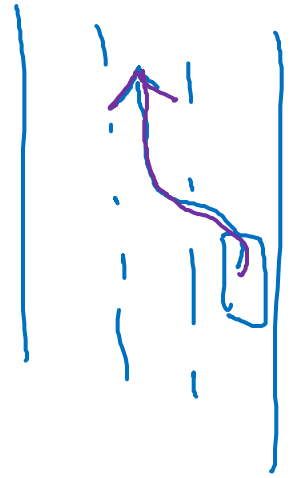
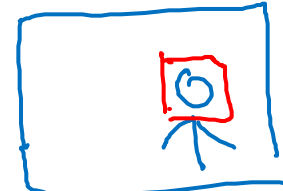
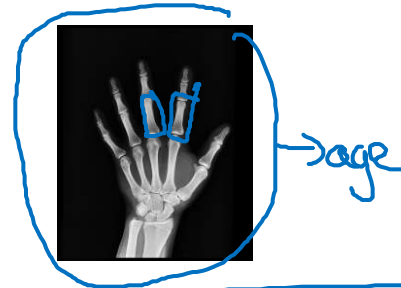
Data.  
- - -

Hand-design.  
- - -

# Applying end-to-end deep learning

Key question: Do you have sufficient data to learn a function of the complexity needed to map  $x$  to  $y$ ?

$x \rightarrow y$



- Use DL to learn individual components
- Carefully choose  $x \rightarrow y$  depending what tasks you can get data for.

$\rightarrow$  image  $\rightarrow$  steering