Error Analysis

Carrying out error analysis

deeplearning.ai

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

"ceiling"

→ 5%
5/100
10%
9.5%

→ 50%
50/100

10%
↓
5%

Andrew Ng

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	8%	43%	61%	12%	

Andrew Ng

deeplearning.ai

Error Analysis

Cleaning up Incorrectly labeled data

Incorrectly labeled examples

x

y   1   0   1   1   0   (1)   1

Training Set.

DL algorithms are quite robust to <u>random errors</u> in the training set.

Systematic errors

Andrew Ng

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	8%	43%	61%	6%	

Overall dev set error — — — — — — 10%

Errors due incorrect labels — — — — 0.6% ←

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

2%
0.6%
1.4%

2.1%   1.9%

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

Andrew Ng

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 *98.1%* right as well as ones it got wrong. *2%*

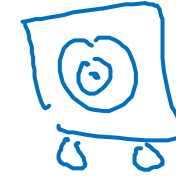- Train and dev/test data may now come from slightly different distributions.

Speech recognition example

- Noisy background
  - Café noise
  - Car noise
- Accented speech
- Far from microphone
- Young children's speech
- Stutter
- ...

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.

Cat app example

Data from webpages

Care about this

Data from mobile app

$\approx 200,000$

$210,000$

$\approx 10,000$

shuffle

X Option 1:

| train | dev | test |

$205,000$

$2,500$
$2,500$

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

$2381 - web$
$119 - mobile$
app

Option 2:

| train | dev | test |
| web | | |

train: $205,000$

app

app
2500

app
2500

Andrew Ng
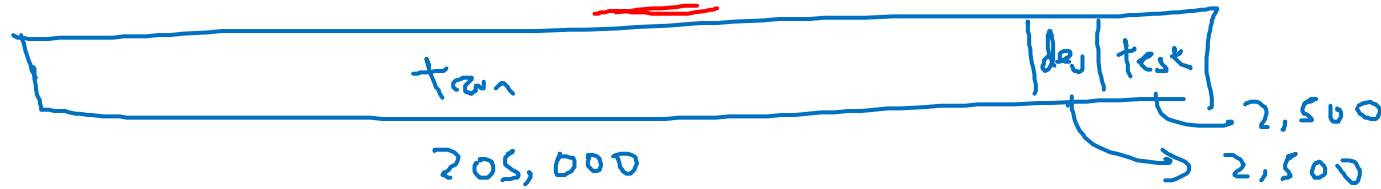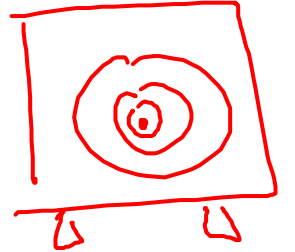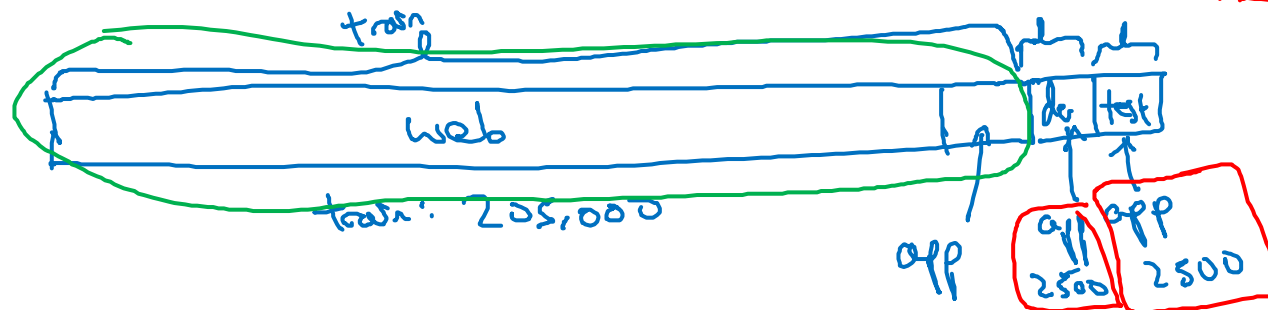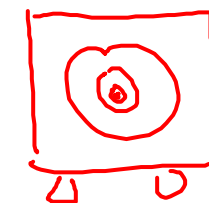
Speech recognition example

Speech activated rearview mirror

Training

Purchased data x, y

Smart speaker control

Voice keyboard

...

500,000 utterances

Dev/test

Speech activated rearview mirror

20,000

10K   5K 5K D T

train
500 K

510K   D T

10K mirror   5K 5K
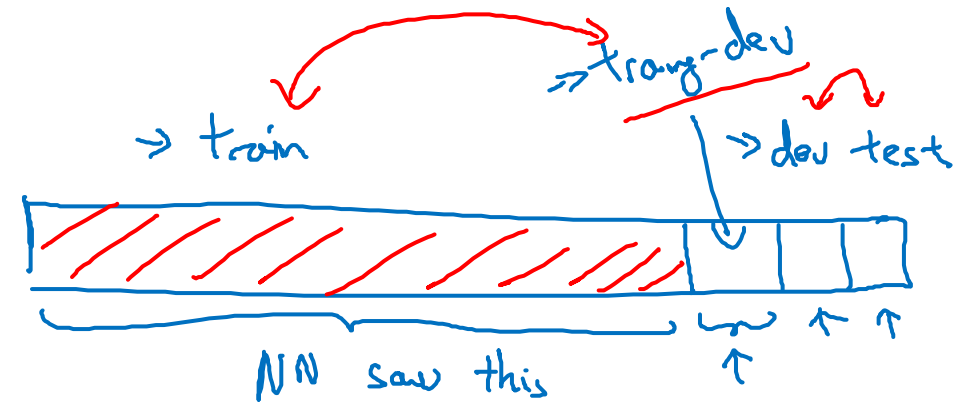
Andrew Ng

Mismatched training
and dev/test data

Bias and Variance with
mismatched data
distributions

Cat classifier example

Assume humans get ≈ 0% error.

Training error 1%

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

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

train → train-dev → dev test

NN saw this

Traing error	1%	1%
→ Traing-dev error	9% ↕ Variance	1.5%
→ Dev error	10%	10% ↑ Data mismatch

Variance

Human error	----- 0% ↕ Avoidable bias	10% ↓ Avoidable bias
Traing error	10%	
Traing-dev error	11%	11% ↕ Variance
Dev error	12%	20% ↑ Data mismatch
	Bias	Bias + Data mismatch

Andrew Ng

Bias/variance on mismatched training and dev/test sets

Human level   4%

Traing set error   7%

Traing - dev set error   10%

→ Dev error   12%

→ Test error   12%

} avoidable bias

} variance

} data mismatch

{ degree of overfitting to dev set.

4%

7% }

10% }

6% }

6% }
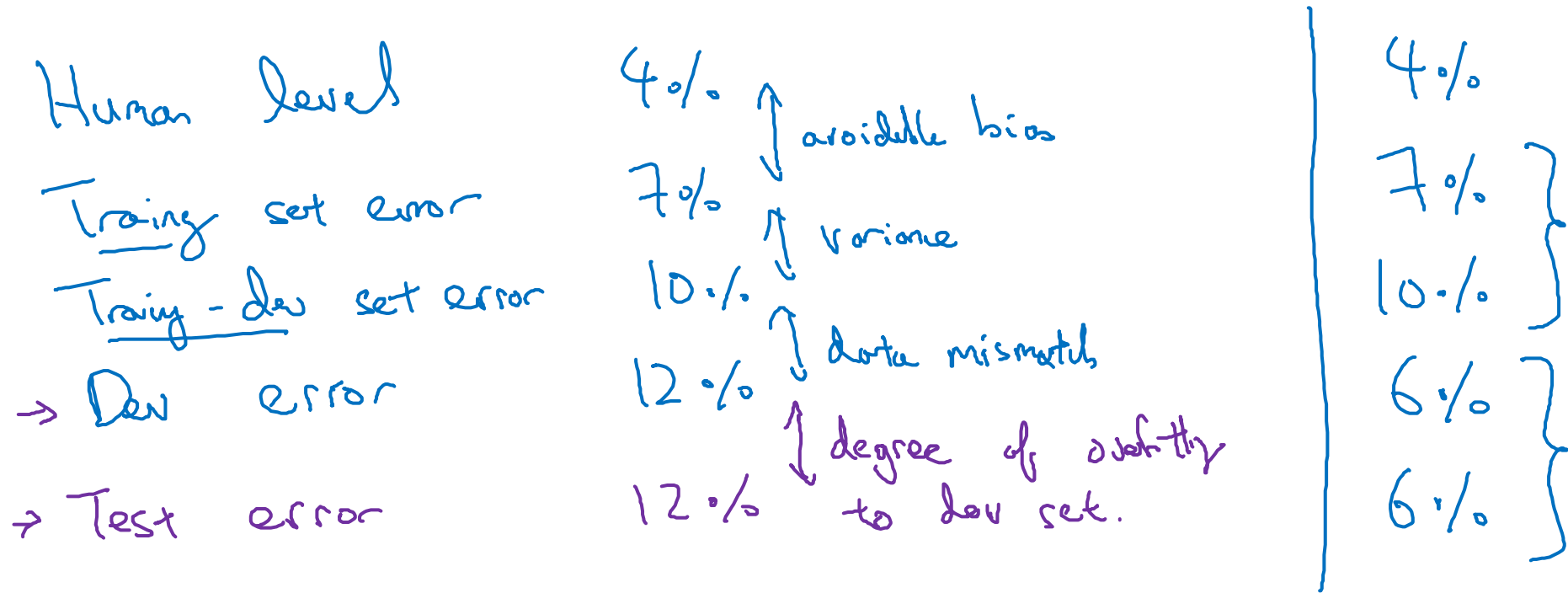
More general formulation

Rearview Mirror

	General speech recognition	Rearview mirror speech data.
Human level	"Human level" 4%	6%
Error on examples trained on	"Training error" 7%	6%
Error on examples not trained on	"Training-dev error" 10%	Dev/Test error 6%

Avoidable bias

Variance

data mismatch

Andrew Ng
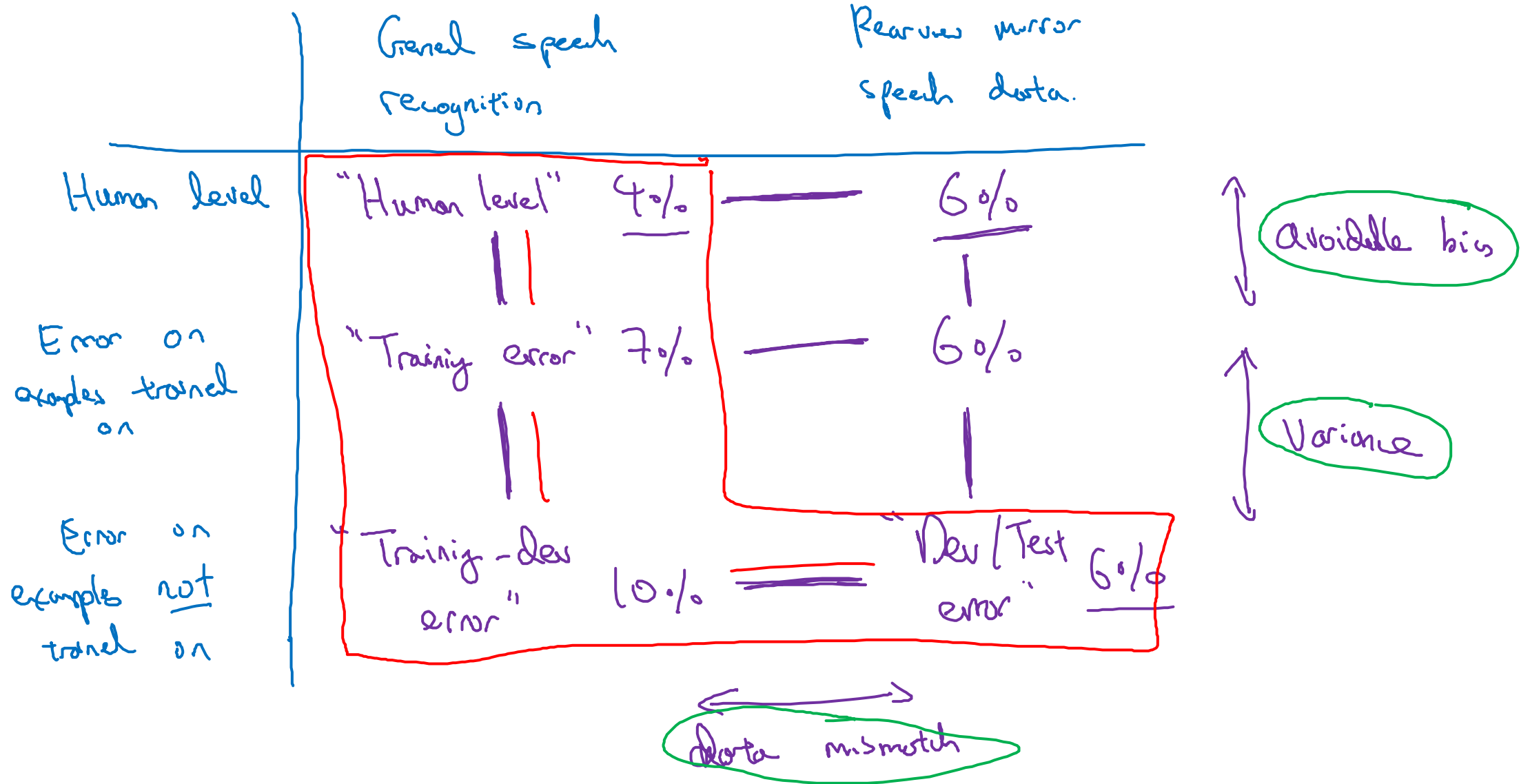
deeplearning.ai

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

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

E.g. Simulate noisy in-car data
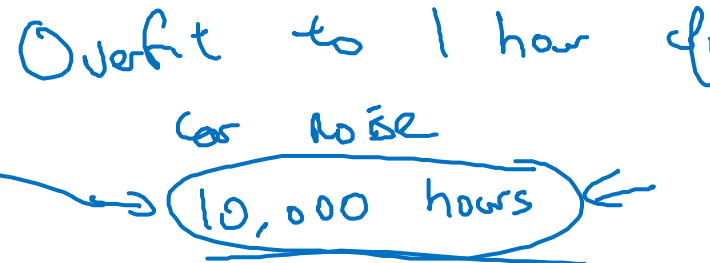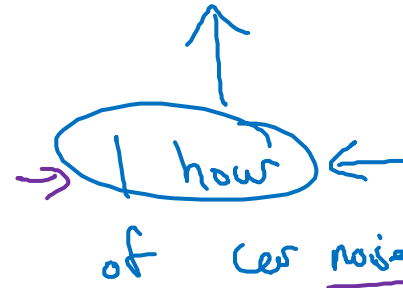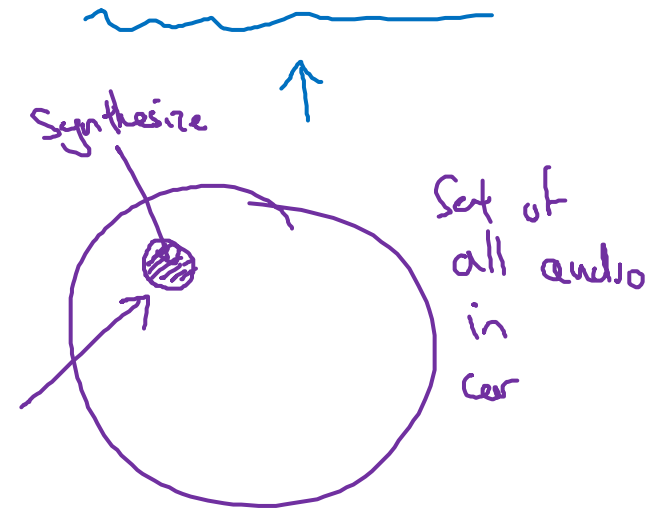
Artificial data synthesis

"The quick brown
fox jumps
over the lazy dog."

+   Car noise   =   Synthesized
in-car audio

10,000 hours

1 hour
of car noise

Overfit to 1 hour of
car noise
10,000 hours

Synthesize

Set of
all audio
in
car

Andrew Ng

Artificial data synthesis

Car recognition:

≈ 20 cars

Synthesized · All cars

Learning from
multiple tasks

Transfer learning

deeplearning.ai

Transfer learning

image recogn. (x,y)

pre-training

$\rightarrow (x,y)$ — fine-tuning

radiology image

diagnoses

image recognition

$\rightarrow \boxed{1,000,000}$ $\boxed{100} \leftarrow$

$w^{[L]}, b^{[L]}$

$\rightarrow \boxed{\text{radiology diagnosis}}$

$\rightarrow \boxed{100}$ $\boxed{1000}$

Speech recognition

10h   10,000h

wakeword/triggerword detection   1h

50h

Andrew Ng
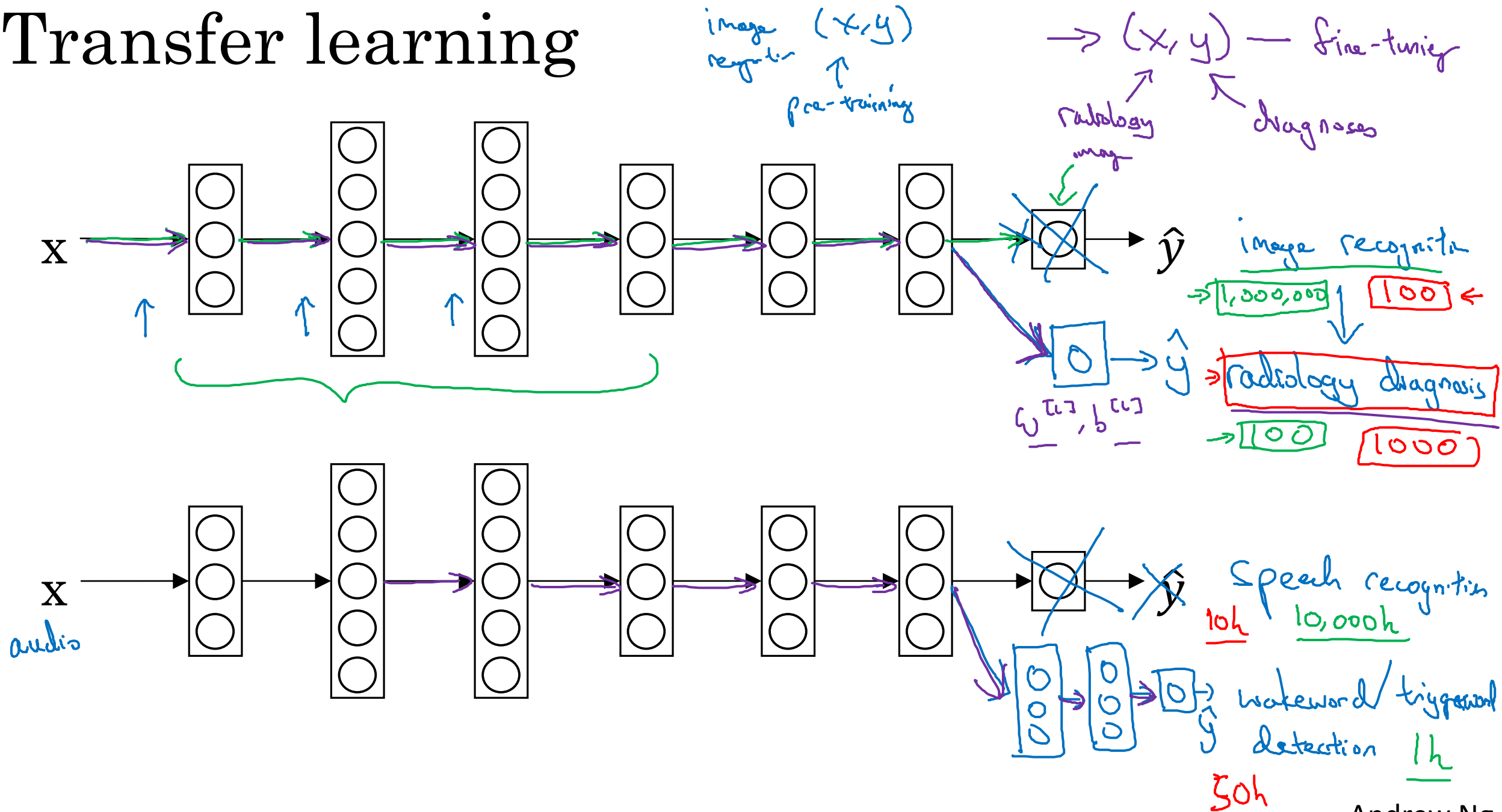
When transfer learning makes sense

Transfer from A → 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
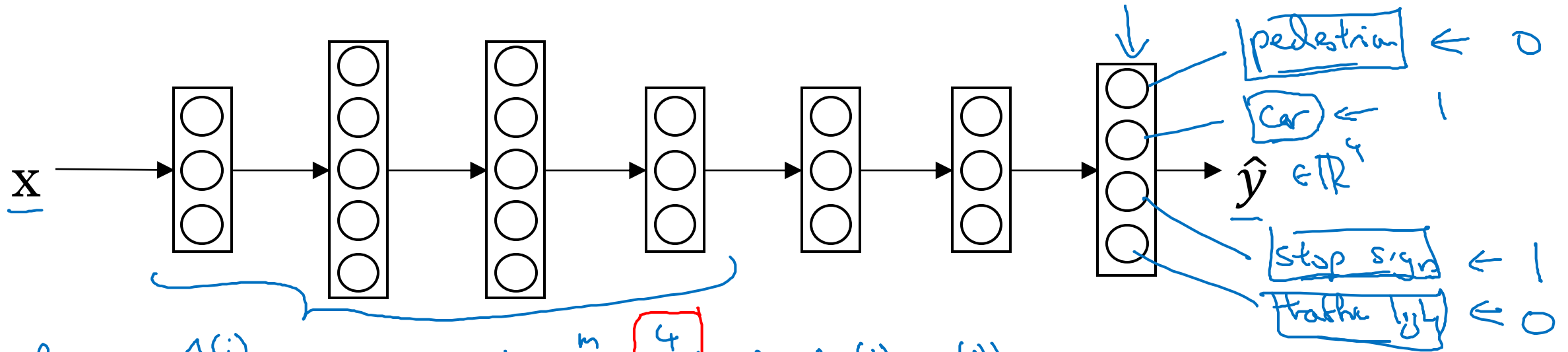
Traffic lights

$\vdots$

$y^{(i)}$

$\begin{matrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \end{matrix}$

$(4, 1)$

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

$(4, m)$

Andrew Ng

Neural network architecture

x → [ ] → [ ] → [ ] → [ ] → [ ] → [ ] → [ ] → $\hat{y}$ $\in \mathbb{R}^4$

pedestrian ← 0
Car ← 1
stop sign ← 1
traffic light ← 0

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

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

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

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

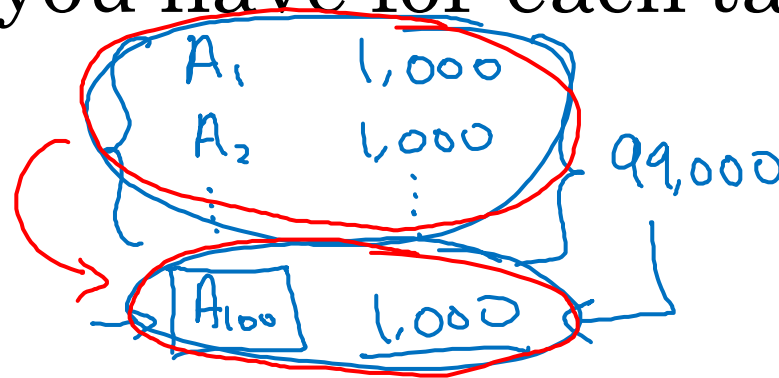Unlike softmax regression:
One image can have multiple labels

Multi-task learning ←

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

Andrew Ng

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.

$$A \quad 1,000,000$$
$$\downarrow \quad \downarrow$$
$$B \quad 1,000$$

$$A_1 \quad 1,000$$
$$A_2 \quad 1,000$$
$$\vdots \quad \vdots \quad 99,000$$
$$A_{100} \quad 1,000$$

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

Andrew Ng

What is end-to-end learning?

Speech recognition example

"cat"

X → audio $\xrightarrow{MFCC}$ features $\xrightarrow{ML}$ Phonemes → Words → transcript Y

audio
→ audio ————————→ phonemes —→ ···· —→ transcript

3,000h
↑

10,000h
⋮
100,000h

Andrew Ng
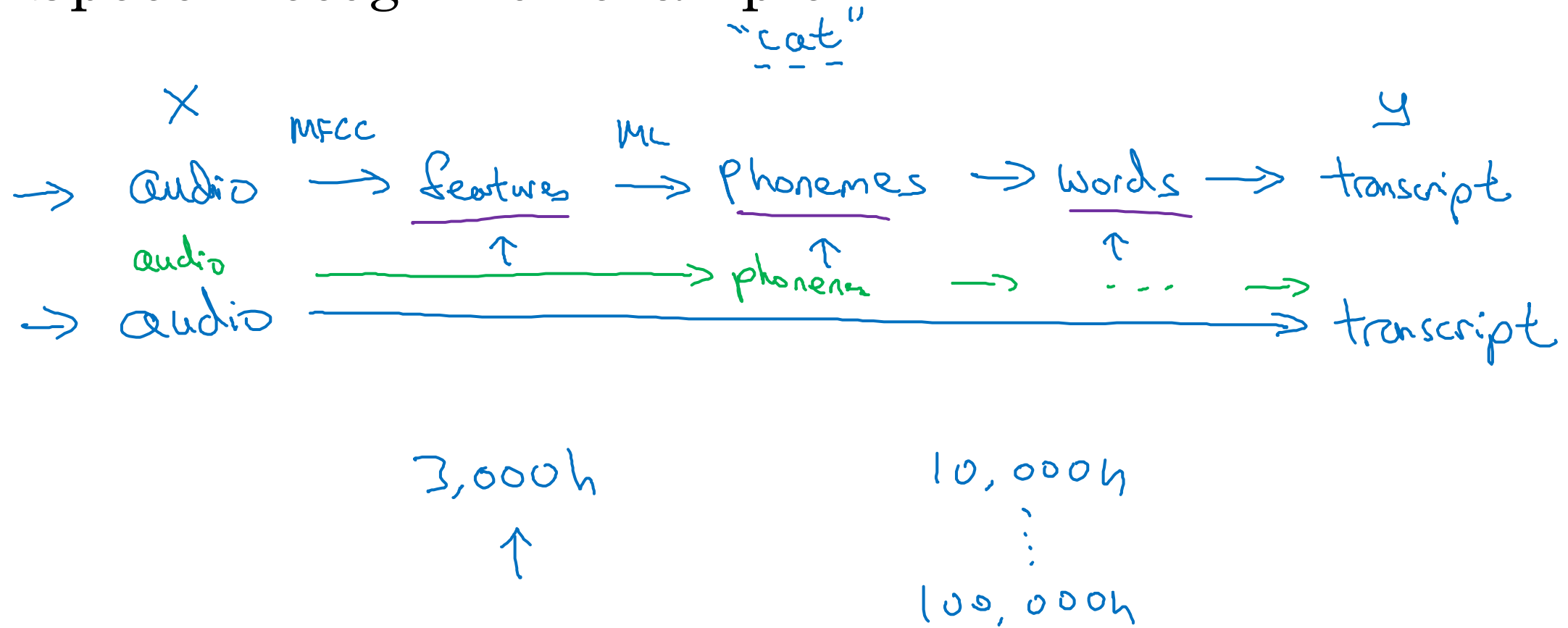
Face recognition

[Image courtesy of Baidu]

Image (x) $\longrightarrow$ y

Identity

(x, y)

① | ②

(x, y)

$\longrightarrow$ Identity

Same person?
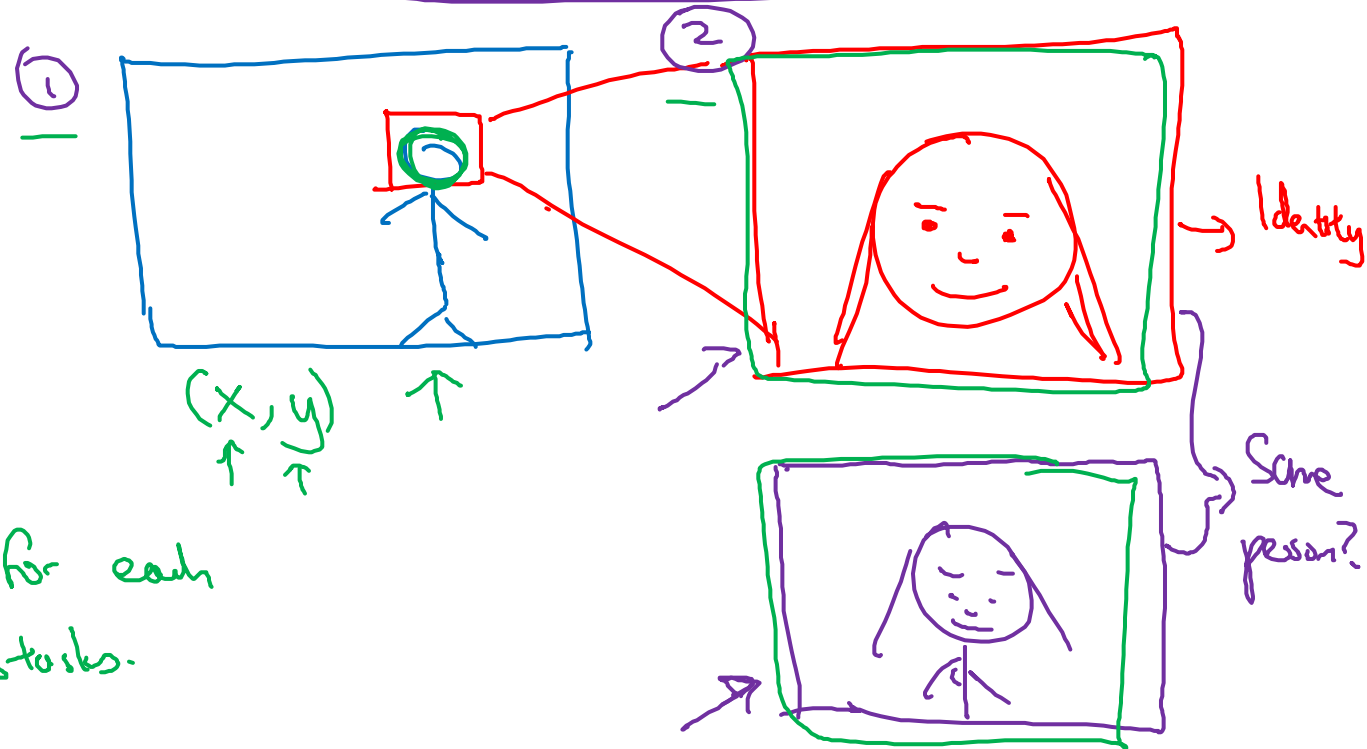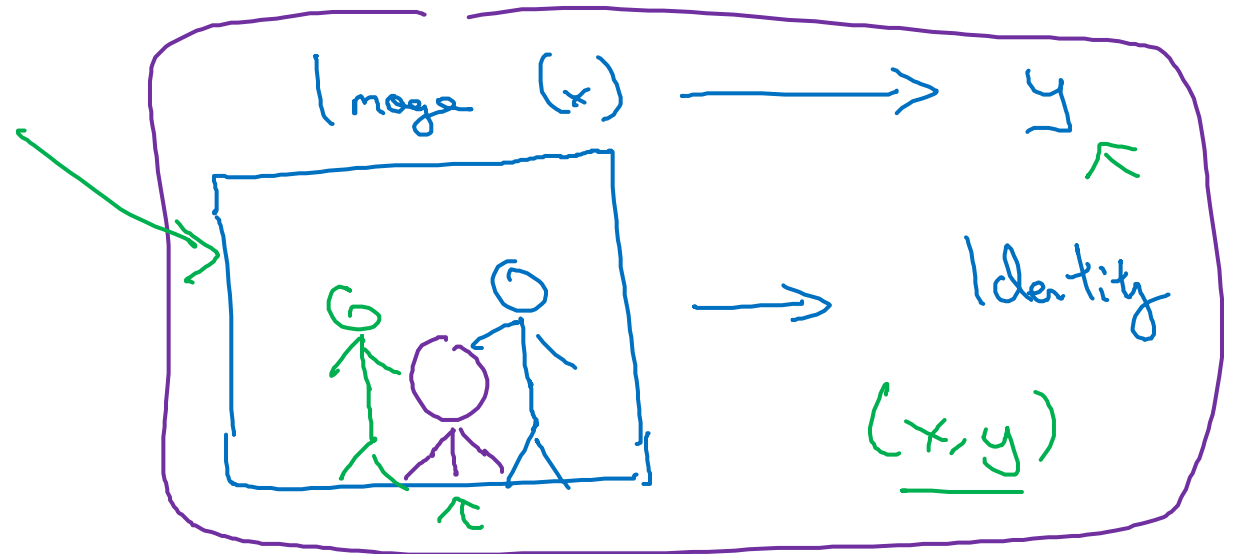
Have data for each of 2 subtasks.

Andrew Ng

More examples

Machine translation

(x, y)

English ↗ ↖ French

English → text analysis → ... → French

English ——————————→ French
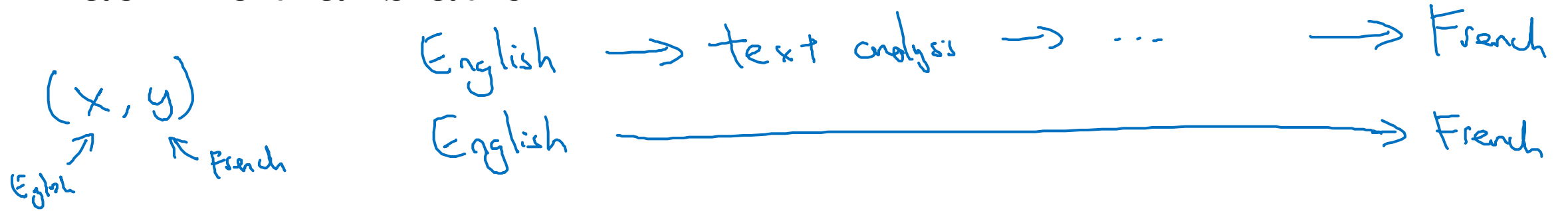
Estimating child's age:

Image ①→ bones ②→ age

Image ——————————→ age ←
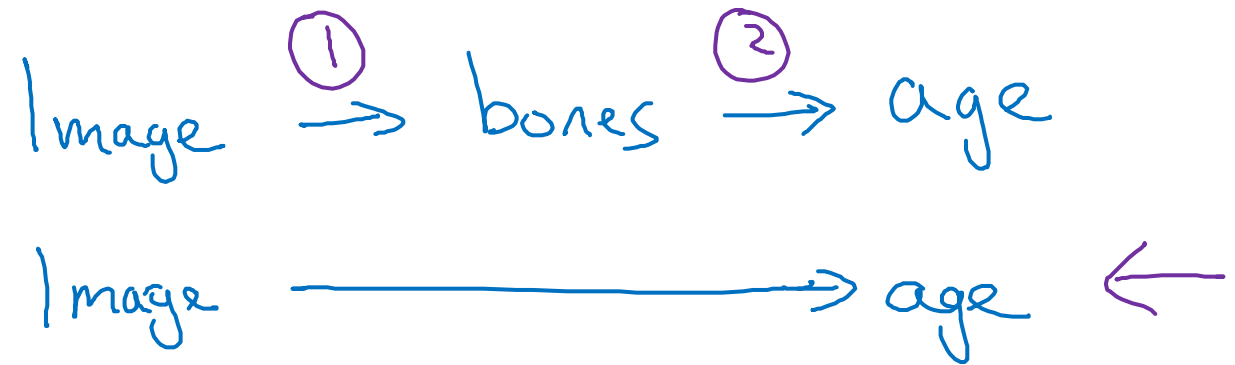
End-to-end deep learning

Whether to use end-to-end learning

Pros and cons of end-to-end deep learning

Pros:

- Let the data speak

$x \longrightarrow y$

$\rightarrow$ "phonemes"

c a t

- Less hand-designing of components needed

$x - - - - - \rightarrow y$

input end

output end

Cons:

- May need large amount of data

$x \longrightarrow y$ (x, y)

- Excludes potentially useful hand-designed components
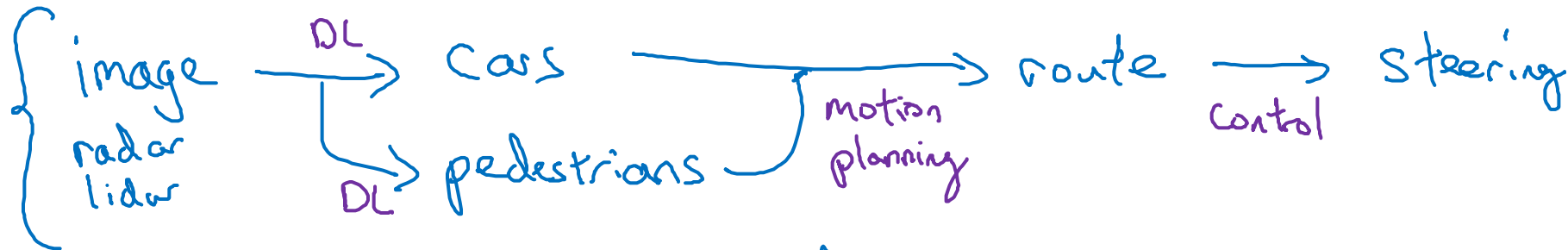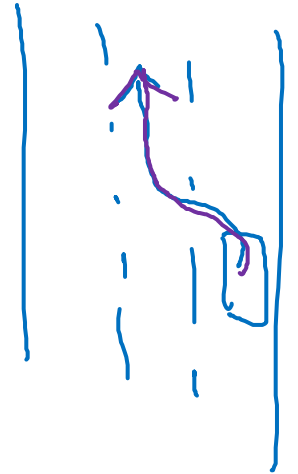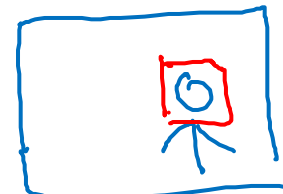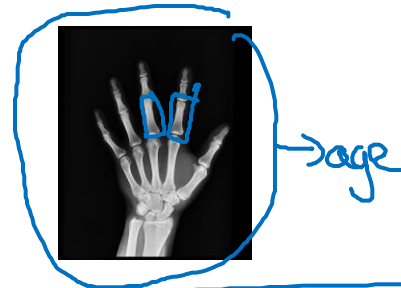
Data.   Hand-design.

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?

$x \rightarrow y$

$\rightarrow$ age

image
radar
lidar
$\xrightarrow{DL}$ Cars
$\xrightarrow{DL}$ pedestrians
motion planning
$\rightarrow$ route
control
$\rightarrow$ steering

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

$\rightarrow$ image $\rightarrow$ steering

Andrew Ng