deeplearning.ai

Introduction to ML strategy

Why ML Strategy?

Motivating example






90%

Ideas:

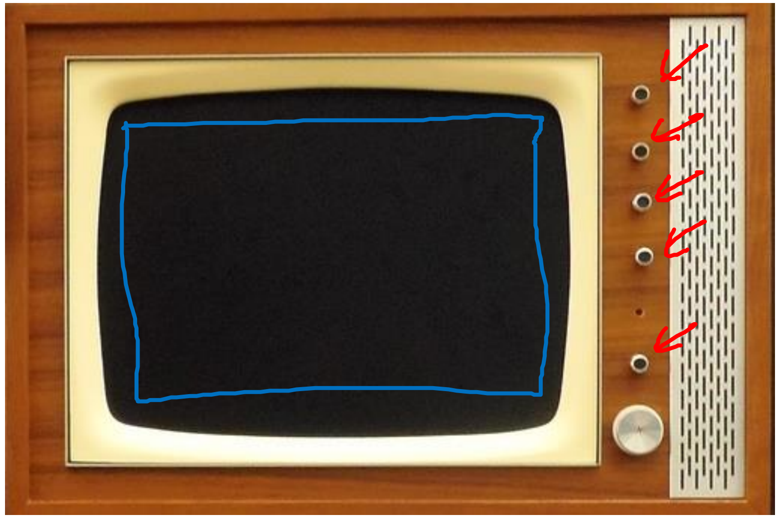
- Collect more data
- Collect more diverse training set
- Train algorithm longer with gradient descent
- Try Adam instead of gradient descent
- Try bigger network
- Try smaller network
- Try dropout
- Add L_2 regularization
- Network architecture
  - Activation functions
  - # hidden units
  - …

deeplearning.ai

Introduction to ML strategy

Orthogonalization

TV tuning example



0.1 x

+ 0.3 x

- 1.7 x

+ 0.8 x

+ ...

Orthogonalization

Car

→ Steering

→ Acceleration
Braking

→ 0.3 x angle − 0.8 speed

→ 2 x angle + 0.9 speed.

speed

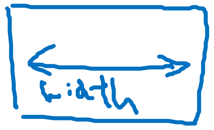
angle

Chain of assumptions in ML

→ Fit training set well on cost function

(≈ human-level performance)

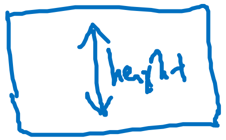
bigger network
Adam
...



early stopping

→ Fit dev set well on cost function

Regularization
Bigger train set



→ Fit test set well on cost function

Bigger dev set

→ Performs well in real world

(Happy cat pic app users.)

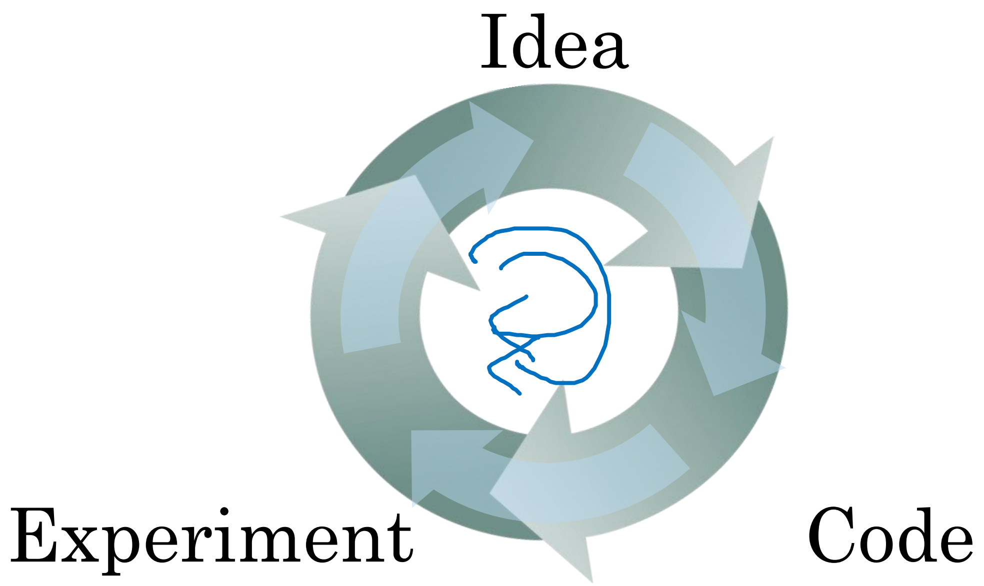
Change dev set or cost function

deeplearning.ai

Setting up your goal

Single number evaluation metric

Using a single number evaluation metric



→ Of examples recognized as cat, what % actually are cats?

→ what % of actual cats are correctly recognized

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

F_1 score = "Average" of P and R.

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

Dev set + Single (real) number evaluation metric

Speed up iterating

Another example

Algorithm	US	China	India	Other
A	3%	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

optimizing

Satisficing

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

cost = accuracy − 0.5 × running Time

maximize accuracy

subject to running Time ≤ 100 ms.

N metrics : 1 optimizing

N-1 satisficing

Wakewords / Trigger words

Alexa, OK Google,

Hey Siri, nihaobaidu

你好百度

accuracy.

#false positive

maximize accuracy.

s.t. ≤ 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:

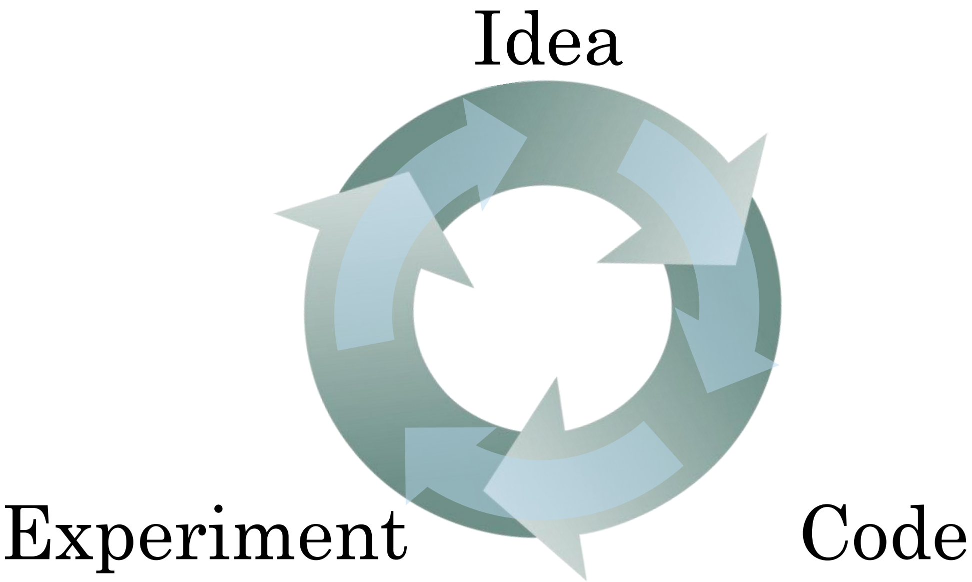
- US
- UK
- Other Europe
- South America
- India
- China
- Other Asia
- Australia

Dev

Test

Randomly shuffle into dev/test

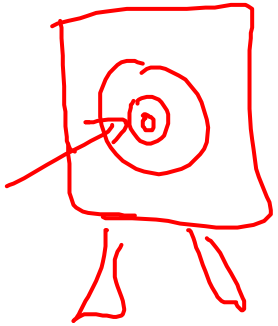
dev set + metric



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

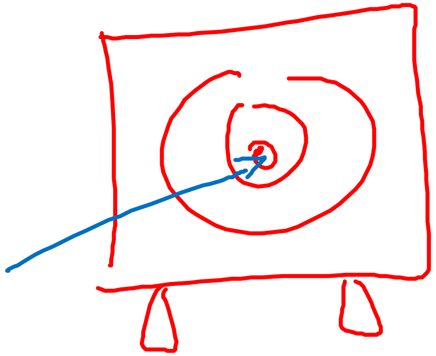
~3 month

Guideline

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.

training



dev
metric

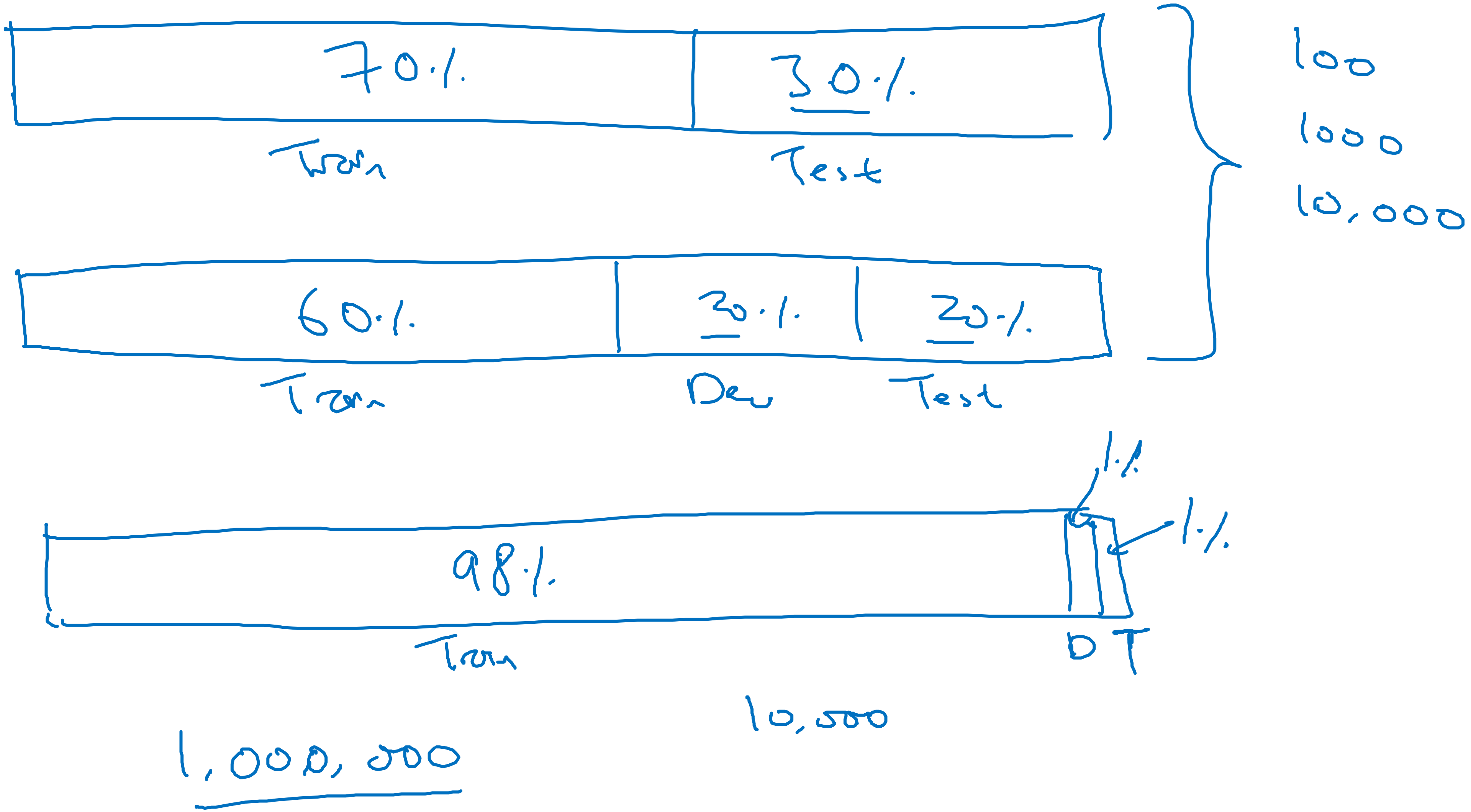
test

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.

A B

97% → 97.1 %

0.1%

100 : small

1%

1,000

10, 000

100, 000

0.01%

0.001%

Online advertising

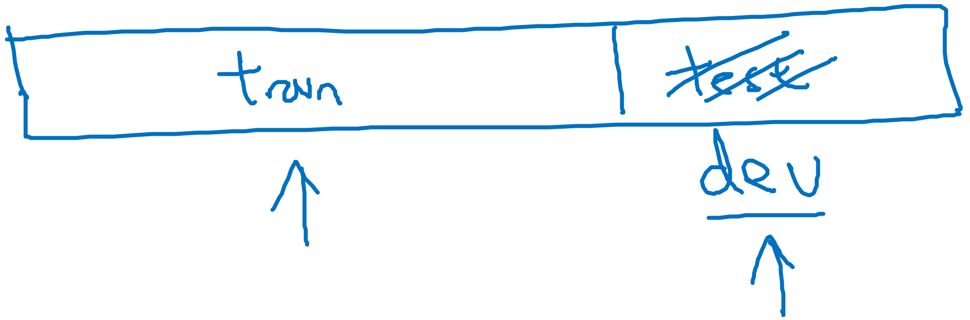
Size of test set

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

10,000

100,000

Train + Dev. No test set.



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

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

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

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

$\sum w^{(i)}$

Another example

Algorithm A: 3% error

Algorithm B: 5% error

Dev/test

User images

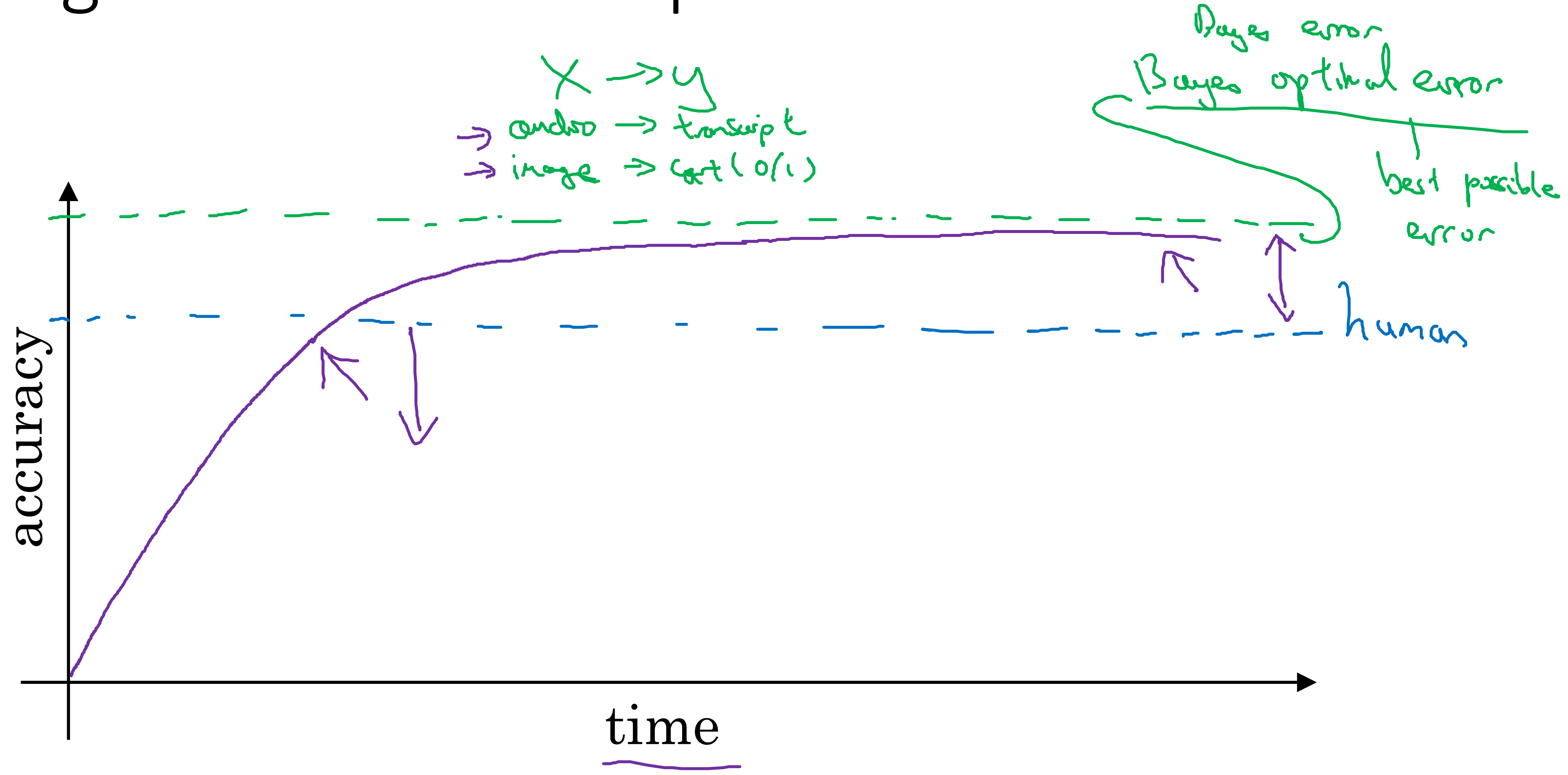
If doing well on your metric + dev/test set does not correspond to doing well on your application, change your metric and/or dev/test set.

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?

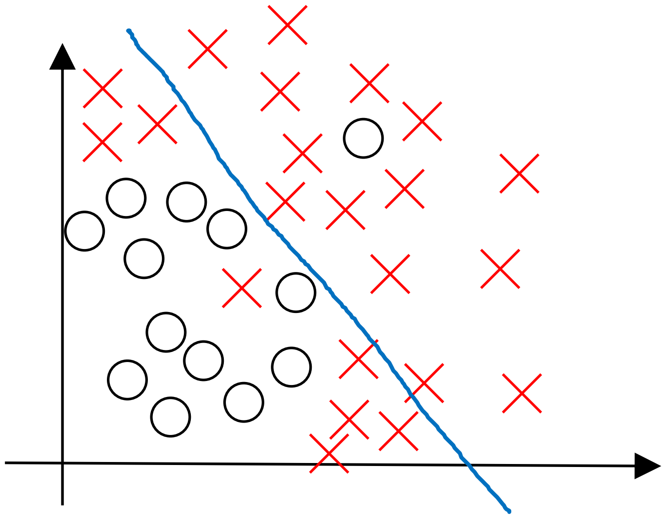
- Better analysis of bias/variance.

deeplearning.ai

Comparing to human-level performance

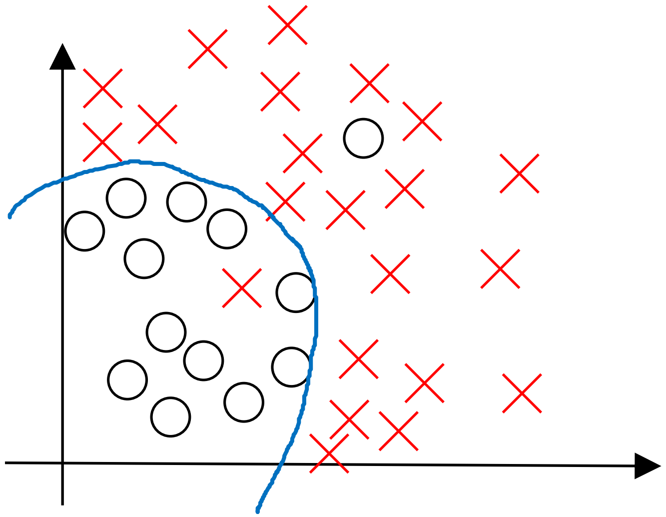
Avoidable bias

Bias and Variance

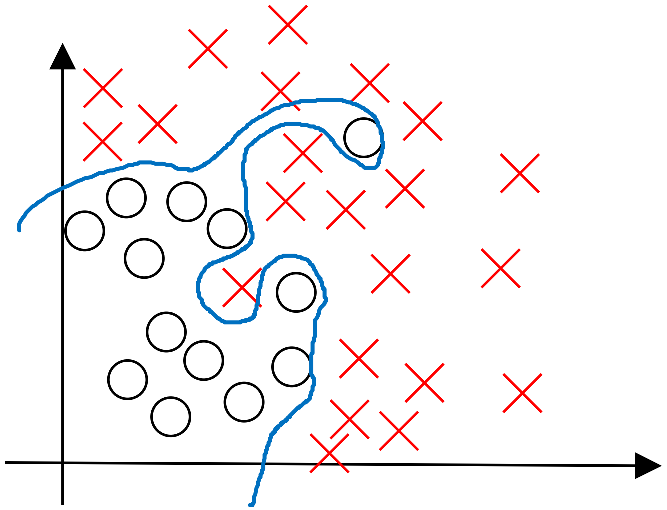


high bias

Underfitting



"just right"



high variance

Overfitting.

Bias and Variance

Cat classification

Human-level ≈ 0% ....

Training set error:

Dev set error:

high variance

high bias

high bias
high variance

low bias
low variance

Cat classification example

Humans (≈ Bayes)	1%	7.5%
Training error	8%	8 %
Dev error	10%	10 %

7%

2%

0.5%

Avoidable bias

2%

2%

variance

Variance

Focus on bias

Focus on variance

Human-level error as a proxy for Bayes error.

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

Baye error ≤ 0.5 %

What is "human-level" error?

Error analysis example

Human (proxy for Bayes error)	1% 0.7% 0.5%	1% 0.7% 0.5%	0.7% 0.5% 1%
Avoidable bias	4% 4.5%	0% 0.5%	0.2% 0.0%
Training error	5%	1%	0.7%
Variance	1%	4%	0.1%
Dev error	6%	5%	0.8%
	Bias	Variance	

Summary of bias/variance with human-level performance

0%

"Bias"

Human-level error

(proxy for Bayes error)

"Avoidable bias"

Training error

"Variance"

Dev error

deeplearning.ai

Comparing to human-level performance

Surpassing human-level performance

Surpassing human-level performance

Team of humans	0.5%	0.5%	0.1%
One human	~~1%~~	1%	0.2%
Training error	0.6%	0.3%	0.3%
Dev error	0.8%	0.4%	

0.1

0.2

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, ...

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

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: → 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%

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%	

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 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	8%	43%	61%	6%	

Overall dev set error 10% | 2%

Errors due incorrect labels 0.6% | 0.6%

Errors due to other causes 9.4% | 1.4%

2.1% 1.9%

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

Correcting incorrect dev/test set examples

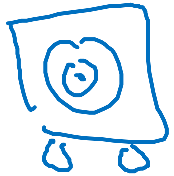
- Apply same process to your dev and test sets to make sure they continue to come from the same distribution
- Consider examining examples your algorithm got 98% right as well as ones it got wrong. 2%
- Train and dev/test data may now come from slightly different distributions.

deeplearning.ai

Error Analysis

Build your first system quickly, then iterate

Speech recognition example



- Noisy background
  - Café noise
  - Car noise
- Accent
- Far fro
- Young
- 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.

deeplearning.ai

Mismatched training and dev/test data

Training and testing on different distributions

Cat app example

care about this

Data from webpages

Data from mobile app

→ ≈200,000 → 210,000 ← ≈10,000

shuffle

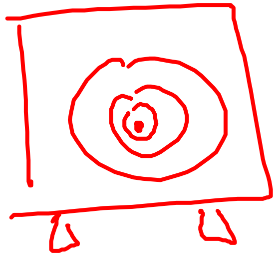
Option 1:

train	dev	test
205,000	2,500	2,500

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

2381 – web

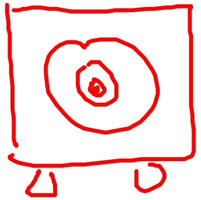
119 – mobile app



Option 2:

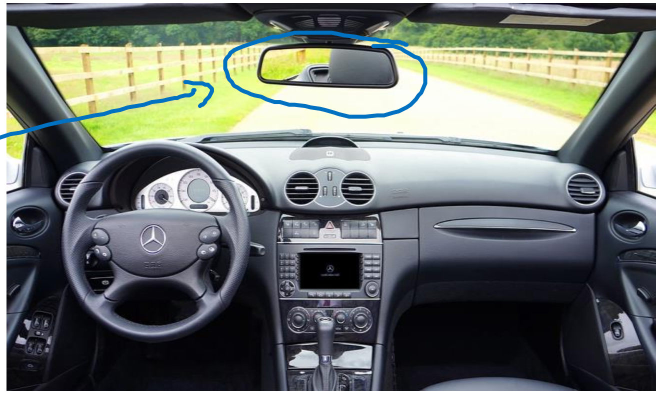
web (train)	dev	test
train: 205,000	app 2500	app 2500

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

→ 20,000

10K 5K 5K

train D T

500K

D T

510K

10K mirror

5K 5K

deeplearning.ai

Mismatched training and dev/test data

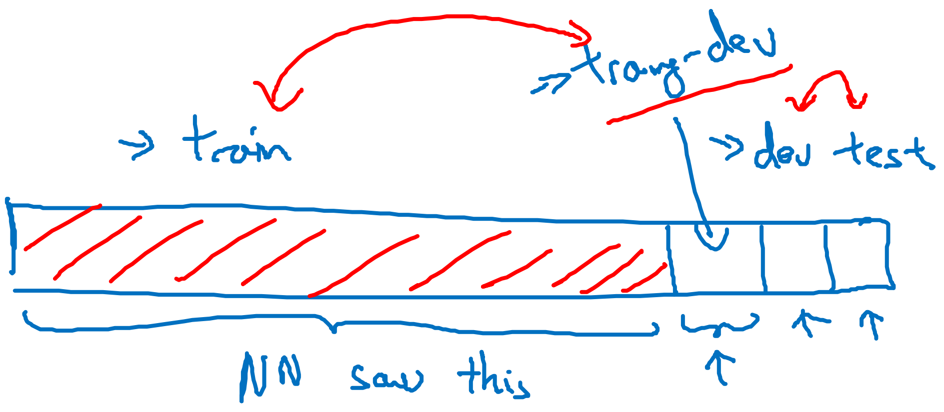
Bias and Variance with mismatched data distributions

Cat classifier example

Assume humans get ≈ 0% error.

Training error 1%
Dev error 10%
9%

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



Training error	1%	1%
Train-dev error	9%	1.5%
Dev error	10%	10%
	Variance	

variance
data mismatch

Human error	0%	
Training error	10%	10%
Train-dev error	11%	11%
Dev error	12%	20%
	Bias	Bias + Data mismatch

Avoidable bias
Avoidable bias
variance
Data mismatch

Bias/variance on mismatched training and dev/test sets

Human level	4%		4%
		avoidable bias	
Training set error	7%		7%
		variance	
Training-dev set error	10%		10%
		data mismatch	
→ Dev error	12%		6%
		degree of overfitting to dev set.	
→ Test error	12%		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 example not trained on	"Training-dev error" 10%	"Dev/Test error" 6%

avoidable bias

variance

data mismatch

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

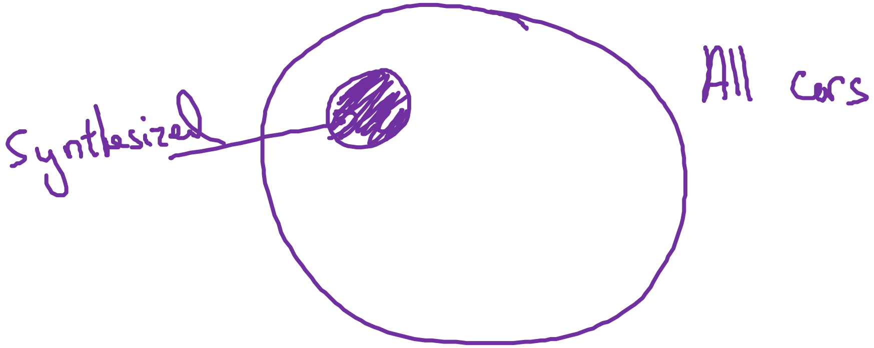
Synthesize

Set of all audio in car

Artificial data synthesis

Car recognition:

≈ 20 cars

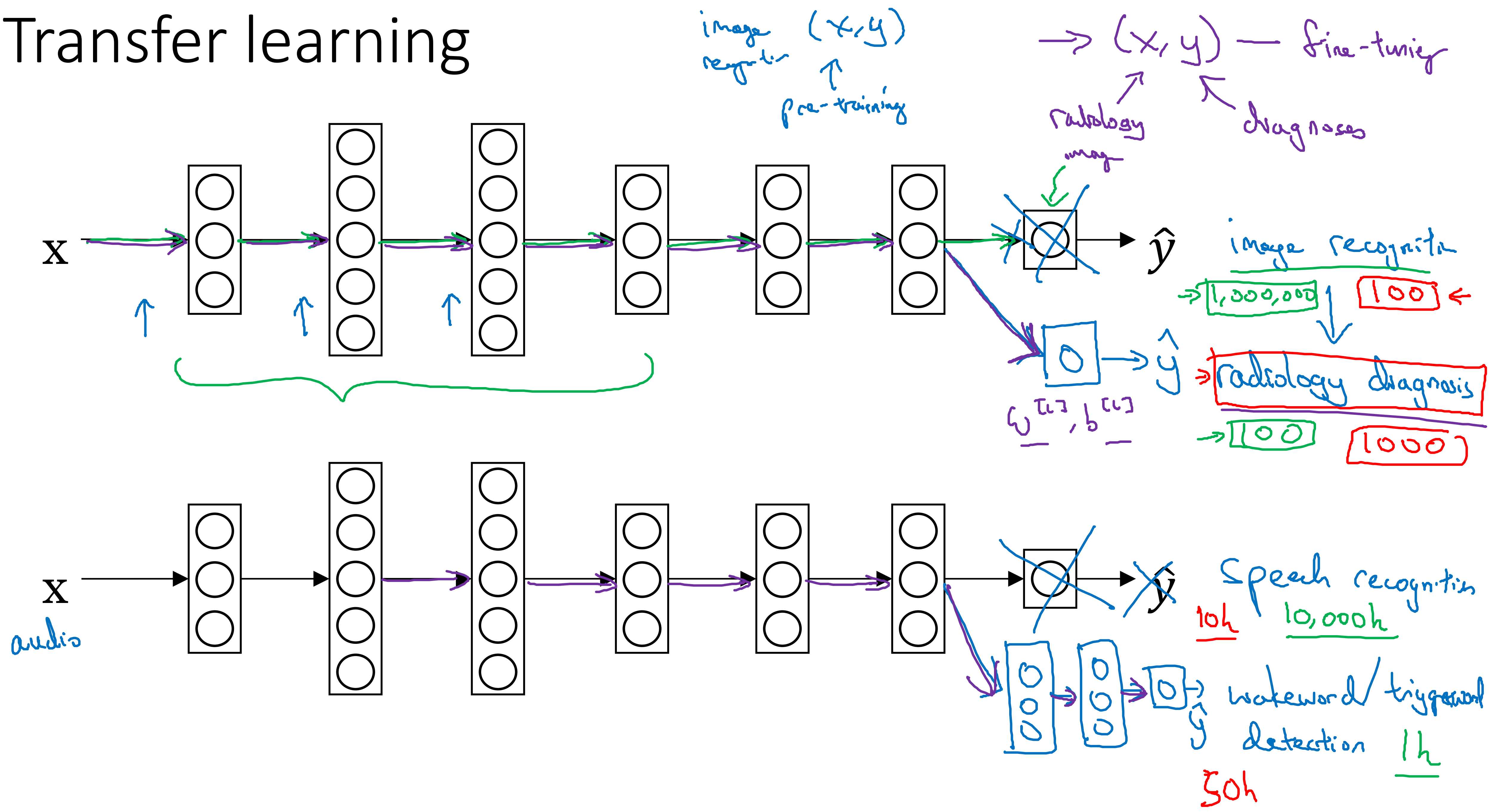


deeplearning.ai

Learning from multiple tasks

Transfer learning

Transfer learning



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.

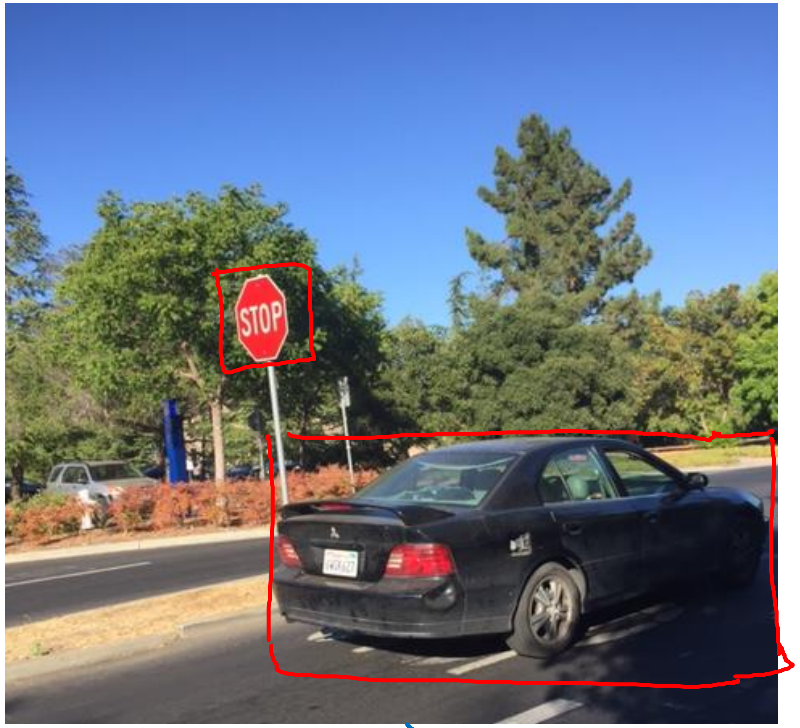
- Low level features from A could be helpful for learning B.

deeplearning.ai

Learning from multiple tasks

Multi-task learning

Simplified autonomous driving example



$x^{(i)}$

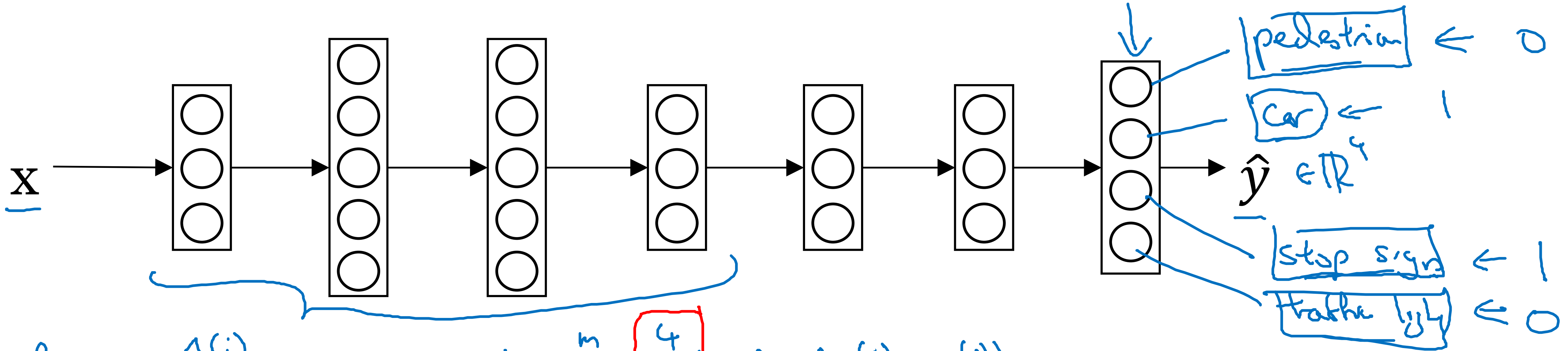
$y^{(i)}$ $(4,1)$

$$\begin{array}{ll} \text{Pedestrians} & 0 \\ \text{Cars} & 1 \\ \text{Stop signs} & 1 \\ \text{Traffic lights} & 0 \\ \vdots & \vdots \end{array}$$

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

$(4, m)$

Neural network architecture



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

Usual logistic loss

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

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

Unlike softmax regression:

One image can have multiple labels

Multi-task learning ←

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

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

A_1 1,000

A_2 1,000

⋮ ⋮

A_{100} 1,000

99,000

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

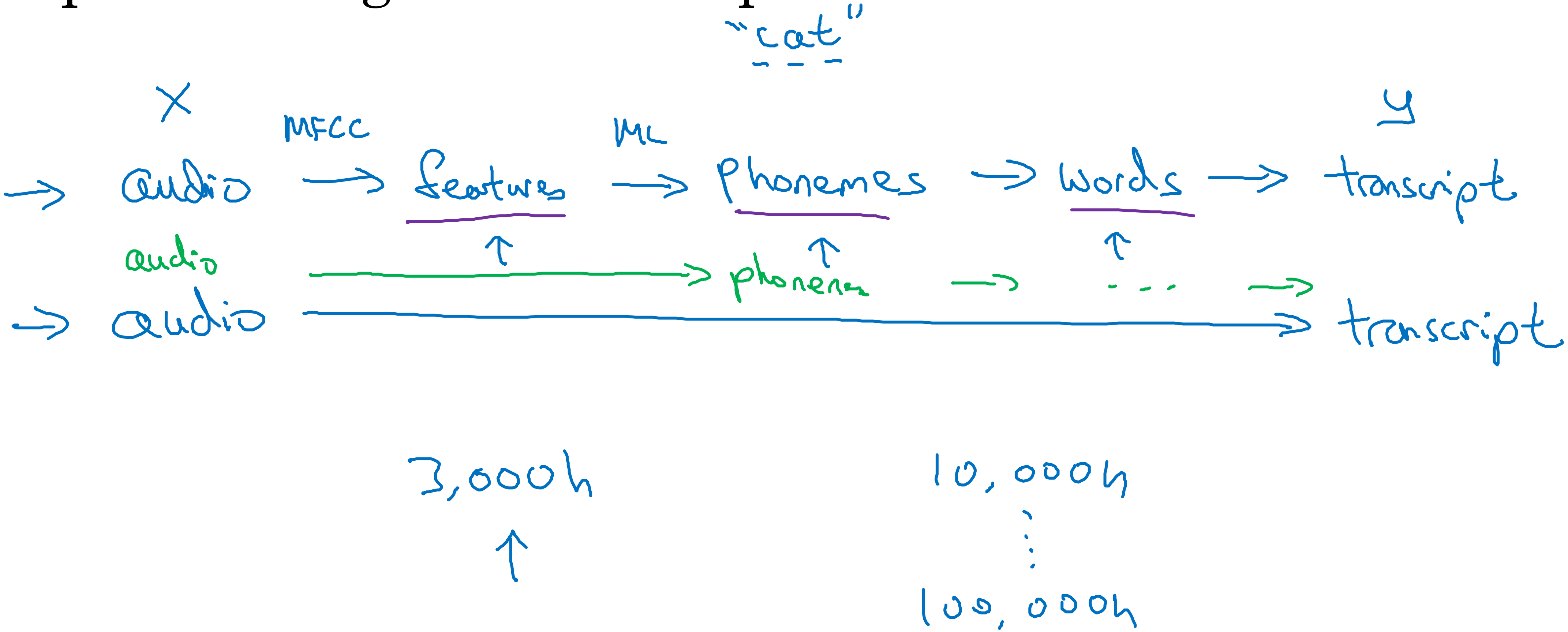
deeplearning.ai

End-to-end deep learning

What is end-to-end deep learning

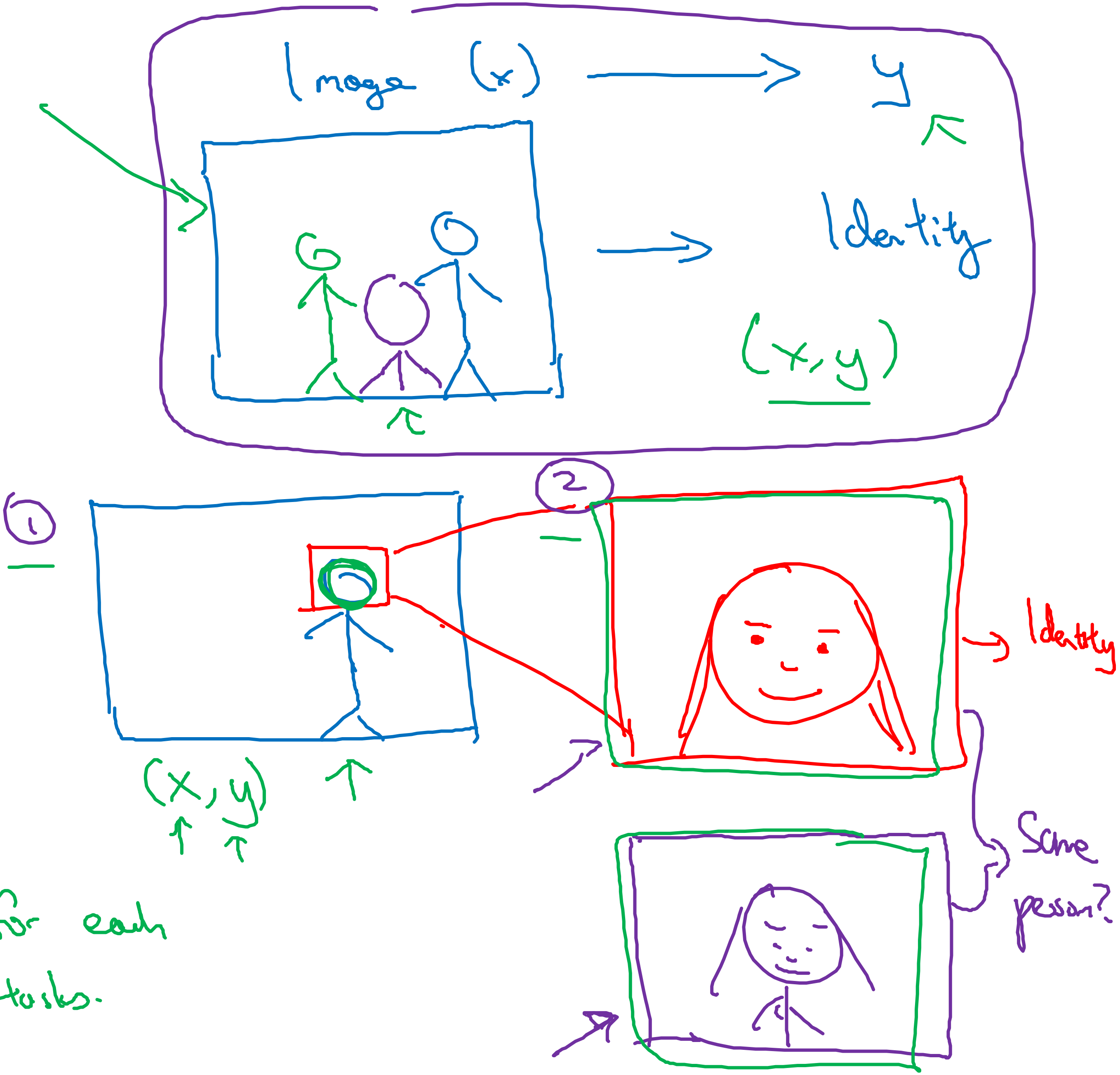
What is end-to-end learning?

Speech recognition example



Face recognition

[Image courtesy of Baidu]



Have data for each of 2 subtasks.

More examples

Machine translation

(x, y)

English French

English → text analysis → ... → French

English → French

Estimating child's age:

Image $\xrightarrow{1}$ bones $\xrightarrow{2}$ age

Image → 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

  $x \longrightarrow y$

  → "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)

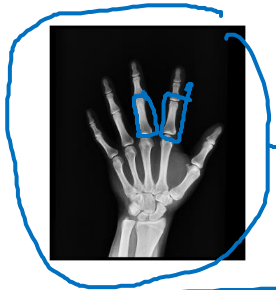
- Excludes potentially useful hand-designed components

  Data. Hand-design.

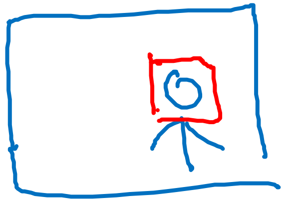
Applying end-to-end deep learning

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

X → y



→ age



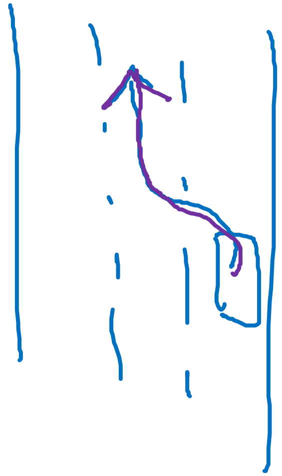


image $\xrightarrow{\text{DL}}$ cars ⟶ route ⟶ steering

radar
lidar

$\xrightarrow{\text{DL}}$ pedestrians

motion planning

control

- Use DL to learn individual components
- Carefully choose X → Y depending what tasks you can get data for.

→ image ⟶ steering