deeplearning.ai

Error Analysis

Carrying out error analysis

Manually examining mistakes can give inside info of what is wrong with it. this process is called 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%

In this case they are relevant and influence a lot.

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%	

Create a table in exel like this and go through the images manually that are not good.
See a errorr % of these categories.

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

you have training set with bad data, incorrect label.

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

Systematic errors

they are lest rubust to systematic errors

if you have wihite dogs consistenly labeled incorrectly.

Andrew Ng

What about incorrect labeled in the dev and test set.

we can fix incorrect labels if they make a significat difference

Error analysis

In this case during error analysis we add one column incorrect label for y

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%	

3 number are worth looking in to decide whether to look into icorrect labeled

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%

so this is big wrt 2 unlike before 200 so in this case its worth going to fix incorrect label.

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

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

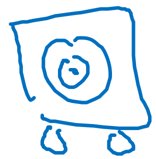
Andrew Ng

deeplearning.ai

Error Analysis

Build your first system quickly, then iterate

Speech recognition example



- Noisy background
  - Café noise
  - Car noise
- Accent
- Far fro
  far from microphone
- Young
  Young children
- Stutter
  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.

Andrew Ng

deeplearning.ai

Mismatched training and dev/test data

Training and testing on different distributions

Training on data with different distrib from dev and test set, here are the problems:

You want your app to do well on the app img

Cat app example

care about this

Data from webpages

Data from mobile app

→ ≈ 200,000

→ ≈ 10,000

210,000

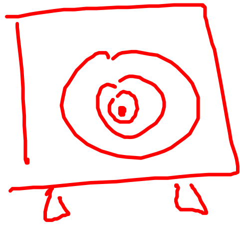
shuffle (singel distrib)

~~Option 1:~~ train | dev | test

205,000

2,500

2,500



$\frac{200k}{210k}$

2381 - web

119 - mobile app

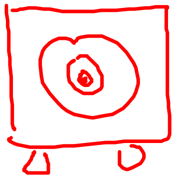
Option 2: train: web | dev | test

train: 205,000

app

app 2500

app 2500



build train with 200000 web+5000 app

for dev and test from app im

Andrew Ng

Say you are building a speech recognizer rearview mirror.
(its real product in China)

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

Andrew Ng

deeplearning.ai

Mismatched training and dev/test data

Bias and Variance with mismatched data distributions

Should you allways use all the data you have???
Estimating the Bias and Variance really helps your learning algorithm really helps you prioritize what to work on next. But the way you analyze bias and variance changes when your training set comes from a different distribution thatn your dev and test sets.

Cat classifier example

Assume humans get ≈ 0% error.

we know bias error is close to 0

Training error	1%	↓ 9%
Dev error	10%	

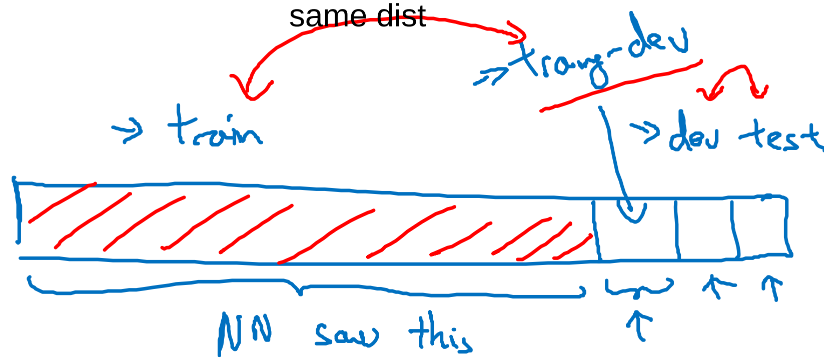
So to carry out error analysis you usually look at the training error and the error on the test set. So say the Training error is 1 % and dev set is 10%, if training and dev come from same distrib you say u have a high variance problem. But when the two come from different distrib you can no longer draw this conclusion. Maybe its just becouse dev set is harder due to blury img

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

: we def this new subset of data

So when you pass from the training set to the dev set, two things change:
1. the alg saw data in training set but not in dev set(variance part of problem)
2. the distrib of data in dev set is different
since u change two things at the same time its diffictult to know of this 9% increase in error how much of it is due to

same dist



we randomly shufle the training set and take a pice train-dev

Now u just train on red part, you dont run backprop on train-dev part

Training error	1%	1%
Training-dev error	9%	1.5%
Dev error	10%	10%
	variance problem	data mismatch

Variance

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

Avoidable bias problem

Avoidable bias / Variance / Data mismatch

Andrew Ng

Bias/variance on mismatched training and dev/test sets

Here is an examples of number that does not allway go up:

Human level	4%	4%
Training set error	7%	7%
Training-dev set error	10%	10%
→ Dev error	12%	6%
→ Test error	12%	6%

avoidable bias

variance

data mismatch

degree of overfitting to dev set.

if this happens that it goes down check next slid

degree of overgitting to dev set
if this happens u find more data into dev set

More general formulation

Rearview mirror

General speech recognition

Rearview mirrorr speech data

	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

Not very much used procedure but sometimes filling this table can help u out

deeplearning.ai

Mismatched training and dev/test data

Addressing data mismatch

There are not systematic way to avoid this problem but there are a few things that we can do:

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

Try to figure out how u dev set is different from test set

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

E.g. Simulate noisy in-car data

Try make training data more similar to training set.
One of the techniques u can use is artificial data sintesis.

Artificial data synthesis

: Make u data look like it has been recorded inside a car

+ =

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

10,000 hours

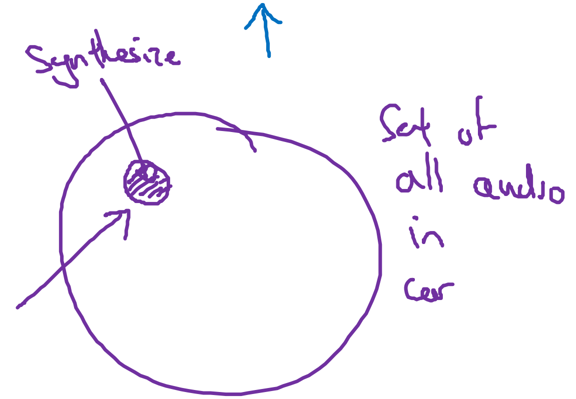
Car noise

1 hour of car noise

Overfit to 1 hour of car noise

10,000 hours

Synthesized in-car audio



Andrew Ng

if u just add hat 1 h of car noise t all 10000 h then
u have a risk to overfit to car noise.
Need to find 10000 h of car noise, hard!!

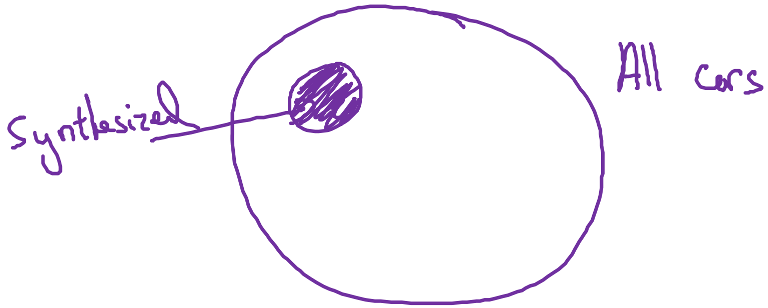
Another example of artif data synthesis

Artificial data synthesis

use computer graphics to generate those img

Car recognition:

≈ 20 cars



Andrew Ng

Artificial data recognition does work in speech recognition

deeplearning.ai

Learning from multiple tasks

Transfer learning

The NN has learned from one task and applying it to other task. U have a NN that recognises cats and then use to do a better job reading x-ray scans This is called transfer learning.

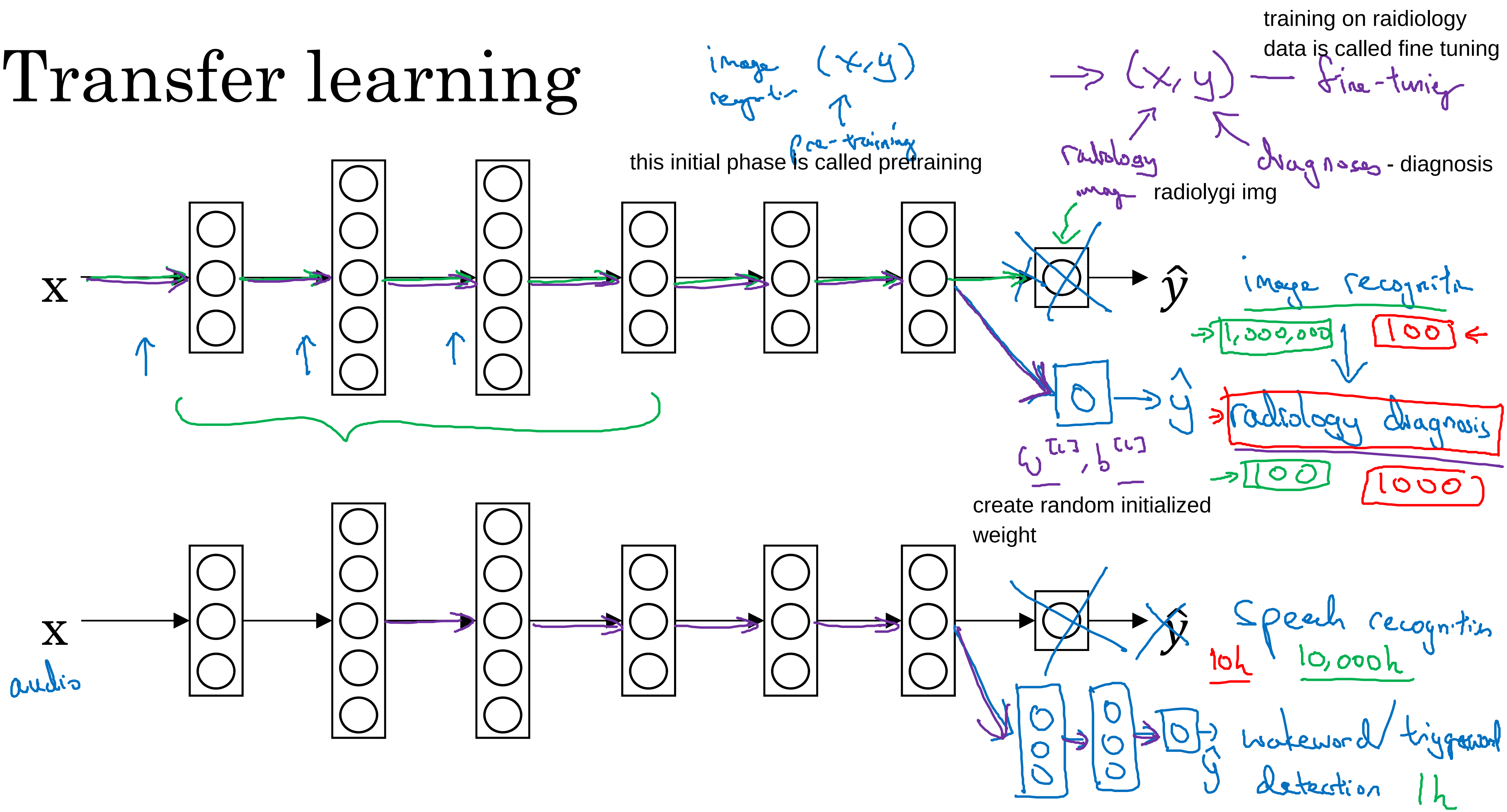
Transfer learning

training on raidiology data is called fine tuning

this initial phase is called pretraining

- diagnosis

radiolygi img



create random initialized weight

We want to build a wakeword or trigger word detection, are words used in our house such as saying ok Google to wake up google, hi Siri ecc.

if you have small training set
just train some layer in the end

Andrew Ng

When does transfer learning make sense?? It makes sense when u have a lot of data for the problem you are transfering from, and usually less data for the problem you're transfering to. It would not make sense if the amount of data u have is inverse.

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

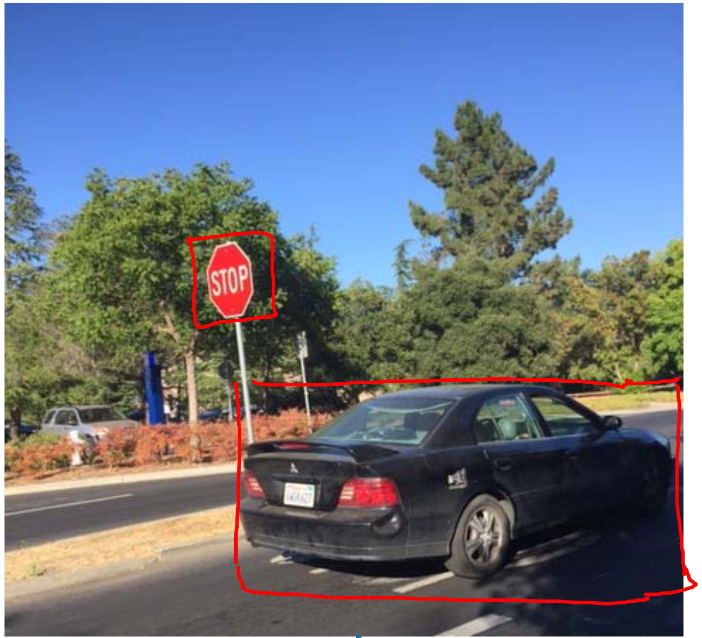
Learning from multiple tasks

Multi-task learning

In multitasking you have one NN trying to do several things at the same time
and each of these tasks helps the other tasks
thats look at an example

Say you are building a self driving car, that has to detect pedestrians, other cars, stop signes, trafic ligjts.ecc

Simplified autonomous driving example



$x^{(i)}$

input x i

$y^{(i)}$ is a vector $(4,1)$

Pedestrians	0
Cars	1
Stop signs	1
Traffic lights	0
⋮	⋮

instead of having one label
u actually have four labours

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

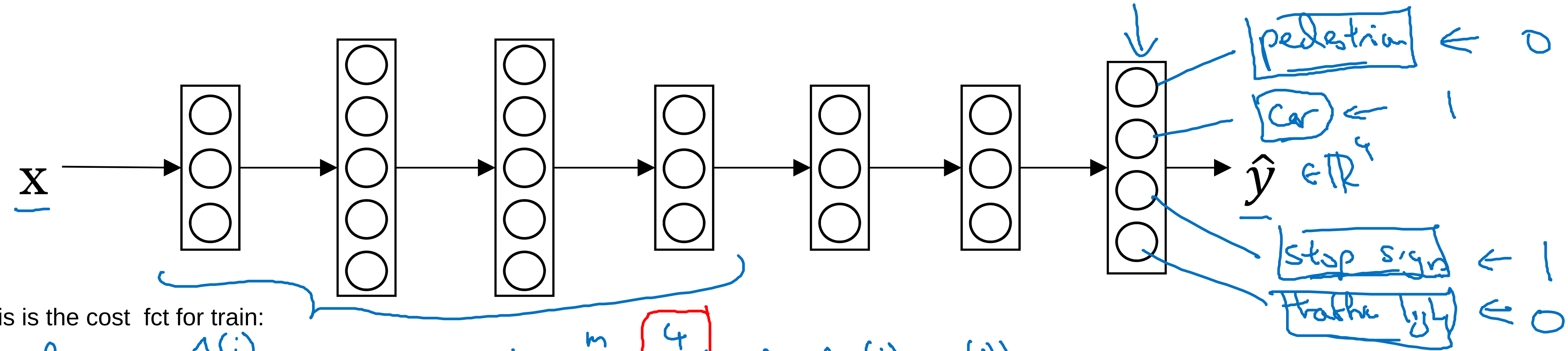
$(4, m)$

So now Y is (4,n) matrix and not a (1,n) matrix like before
So what you can do is to train a NN to predict these values Y

Andrew Ng

Neural network architecture

You can do multiclassification



This is the cost fct for train:

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

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

→ 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

One img can have multiple labels

Multi-task learning ←

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

← when you have some unlabeled, with ?, in the summation u ommit this and just consider sum over 0/1 label

Andrew Ng

If we do min that cost function we are performing Multiclass classification learning

You could have done four separate NN to do this, but like this you get better performance

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

transfer learning

A_1 1,000
A_2 1,000
⋮ ⋮
A_{100} 1,000

99,000

multitask learning

its better as you have for each task less data then when u consider together

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

The only time when multitask learning does hurt performace is when the NN is not big enough.

transfer learning is used much more today then multi-task learning
In computer vision multitask is used much more and in computer detection.

deeplearning.ai

End-to-end deep learning

What is end-to-end deep learning

One of the most recent developements in deep learning has been the rise of end to end learning

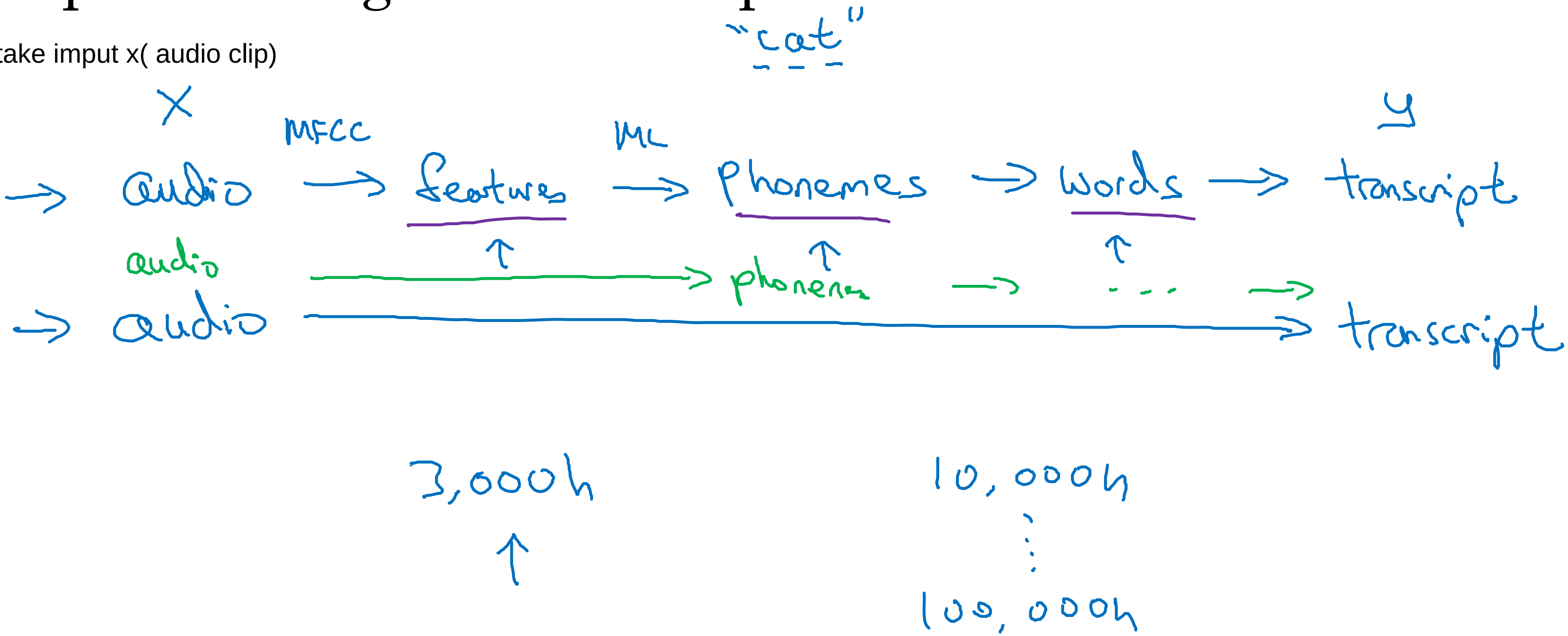
There have been some learning systems that require multipe stages of processing, what end to end does is to take all of these stages an replace with a single NN

What is end-to-end learning?

thats take speech recogn ex:

Speech recognition example

take imput x(audio clip)

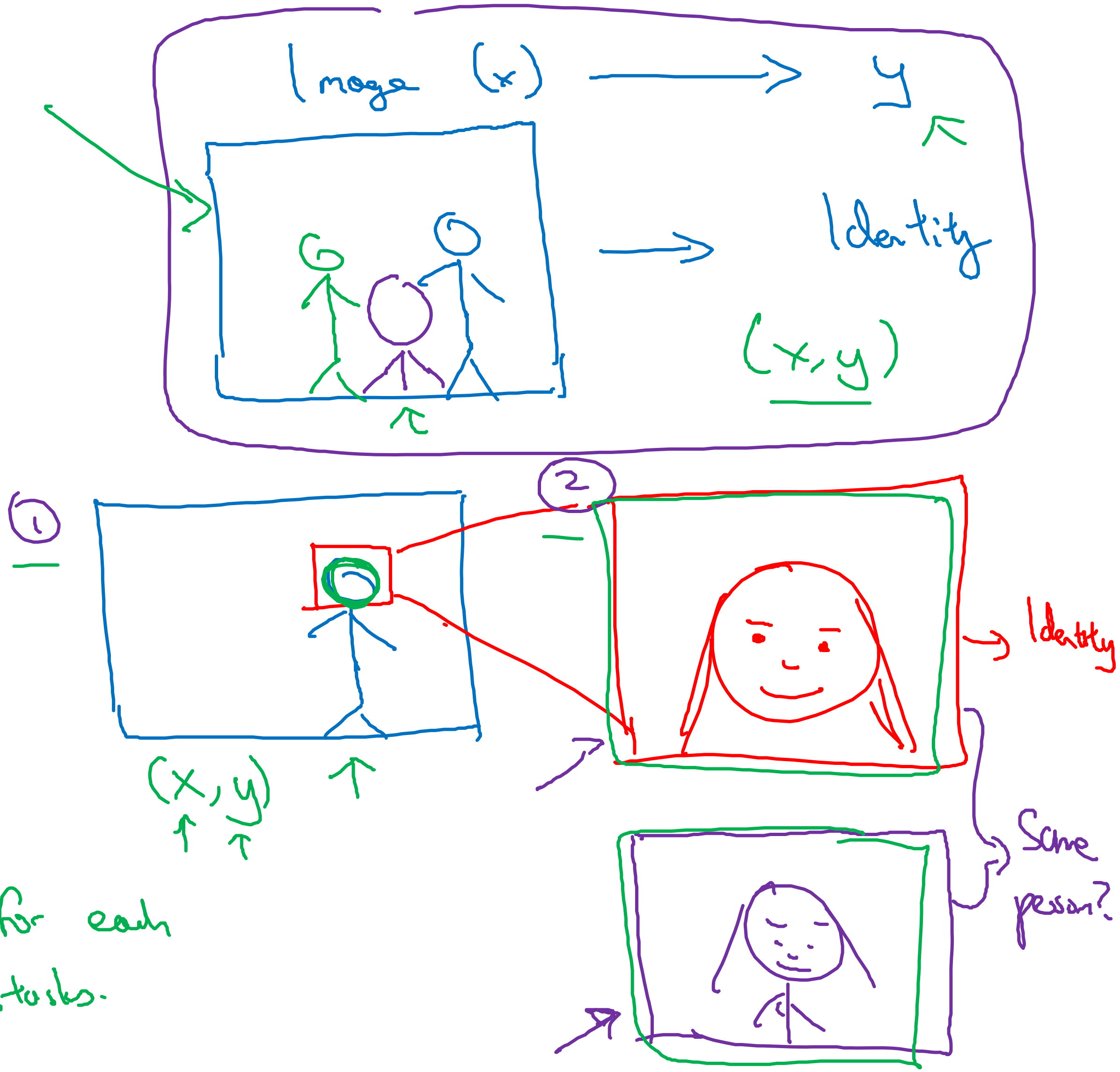


speech recognition involves several phases like using MFCC extract som low level features then use ML to find phonemes(which are the basic units of sound, so for ex word cat is made of three sounds c a t. Then you string together phonemes to form individual words and then you string those together to form the transcripts of the audio clips.

So in contrast to the pipeline with a lot of stages what end to end deep learning does is you can train a huge NN to just input the audio clip and have directly the trascript. The challenge is that you need a lot of data for it to work well, so for ex if you are training on 3000 h of data then the traditional pipeline works really well, its only when you have like 10000 h to 100000 h of data then end to end learning works well. There are also intermediate approaches where you go from audio to phonemes to other stages and then to transcript.

Face recognition

[Image courtesy of Baidu]



Have data for each of 2 sub-tasks.

Andrew Ng

you do multitask, first you identify the person then zoom into the face and crop the face pic and feed to NN

So why the two step works better:

1. each of the two problems you're solving is actually much simpler

2. you have a lot of data for each of the two sub-tasks. In particular ther is a lot of data you can obtain for phase detection

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 ←

Andrew Ng

end to end does not allways work thats see when to use it.

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

$x \rightarrow y$

"phonemes"

c a t

- Less hand-designing of components needed

$x - - \quad - \quad - \rightarrow y$

input end

output end

Cons:

- May need large amount of data

$x \rightarrow y$ (x, y)

- Excludes potentially useful hand-designed components

Data. Hand-design.

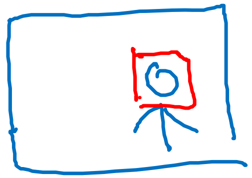
Andrew Ng

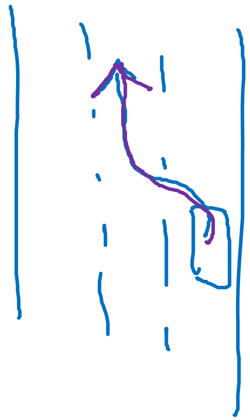
Applying end-to-end deep learning

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

$X \to y$

$\to$ age





image, radar, lidar $\xrightarrow{DL}$ cars $\xrightarrow{\text{motion planning}}$ route $\xrightarrow{\text{control}}$ steering

image, radar, lidar $\xrightarrow{DL}$ pedestrians $\xrightarrow{\text{motion planning}}$ route

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

$\to$ image $\longrightarrow$ steering

not an approach for self driving car

Andrew Ng

You need to be mindful where u apply end to end deep learning