

Structuring Machine Learning Week 2

Carrying out error analysis

When you realize your cat classifier mislabels small dogs as a cats,

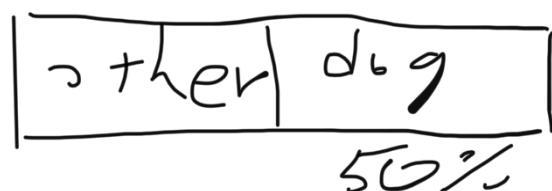
Should you try to make your cat classifier do better on dogs?

Error analysis

- Get ~100 mislabeled dev set examples
- Count Up how many are dogs?



If the dog error rate is relatively small, dealing with the dog problem is not so much effective.



On the other hand, the ratio of that problem is quite big, you could be much more optimistic about spending time on the dog problem.

2020年5月20日 水曜日

Evaluate Multiple ideas on parallel

Image	Dog	Big cats	Blurry	Comments
1	✓			Tiwawa
2		✓		
3		✓	✓	
...				
%	5%	20%	18%	

- Count Up each problems
- Calculate percentage of each problems
- Decide which problem you should pursue

Carrying up incorrectly labeled data

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.

enough and the actual percentage of errors is maybe not too high.

Andrew Ng

- When the labels on the training set has error
- Neural Network is pretty robust against these error
- If you have enough big data set and error ratio is not too high, you don't have to care about this.

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

Training and testing on different distributions

Cat app example

Data from webpages

Data from mobile app

care about this

$\rightarrow \approx 200,000$

$210,000$ (shuttle)

$\approx 10,000$

~~X~~ Option 1:

205,000

2,500

2,500

200K
210K

2381 - web
119 - mobile app

Option 2:

205,000

2,500

2,500

205,000

2,500

2381 - web
119 - mobile app

Andrew Ng

Save Note Discuss Download

When there are two datasets comes from different distributions , and the one which you care about has small number of data. Use them as a dev / test set , And use part of them as training set.

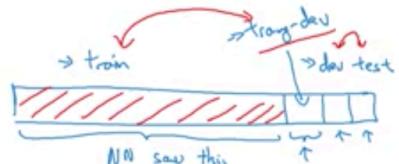
Bias and Variance with mismatched data distributions

Cat classifier example

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

Training error 1% ↓
Dev error 10%

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



Training error 1%
→ Training-dev error 9%
→ Dev error 10%
Variance 1%
Data mismatch 1.5%
10% → 1.5% = 10% data mismatch

Andrew Ng

Save Note

Discuss

Download

	Variance	Data mismatch	Avoidable bias
Human error	1%	1%	0%
Training error	1%	1%	10%
Training-dev error	9%	1.5%	11%
Dev error	10%	10%	12%

Use Training-dev set to distinguish which variance or data mismatch problem you have.

Addressing data mismatch

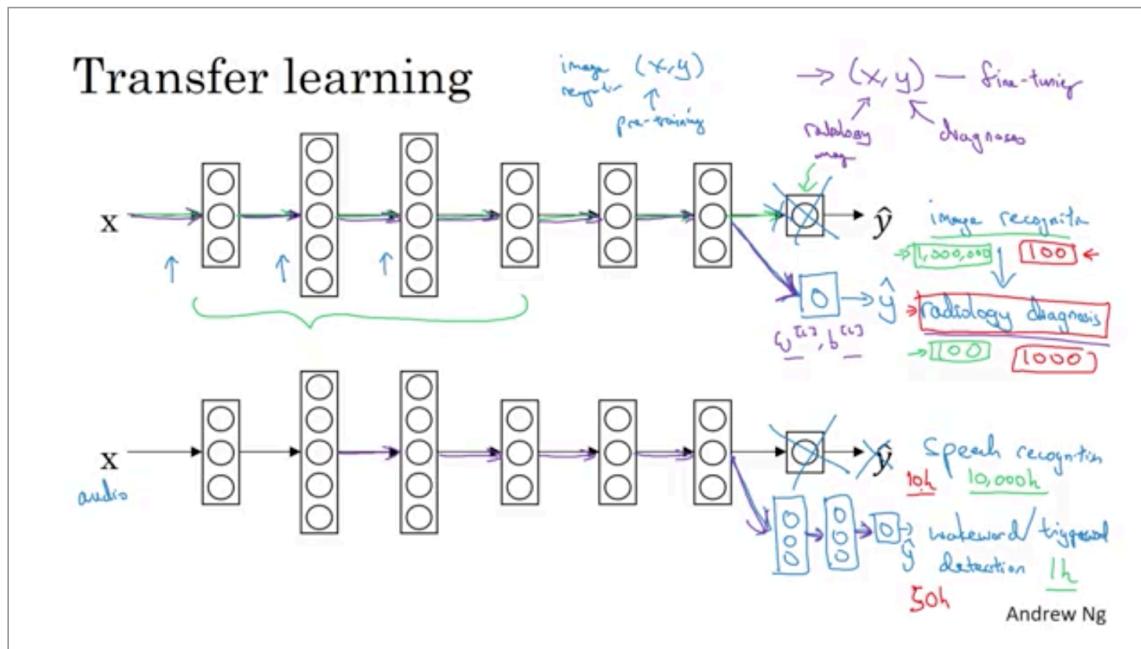
If you think you have a data mismatch problem,

- do error analysis
- or look at training / dev / test set try gain insight into how these two data distributions might differ
- Try to get training data set more looks like dev set
- One recommended way is “Artificial data synthesis”

$$\text{Data} + \text{noise} = \text{Synthetic data}$$

Transfer learning

Transfer learning



Use model that learned from one task to another task.

It can be used when …

- You can't get enough data set (such as reading x-ray scans)
- You might get related but different task

Multi task learning

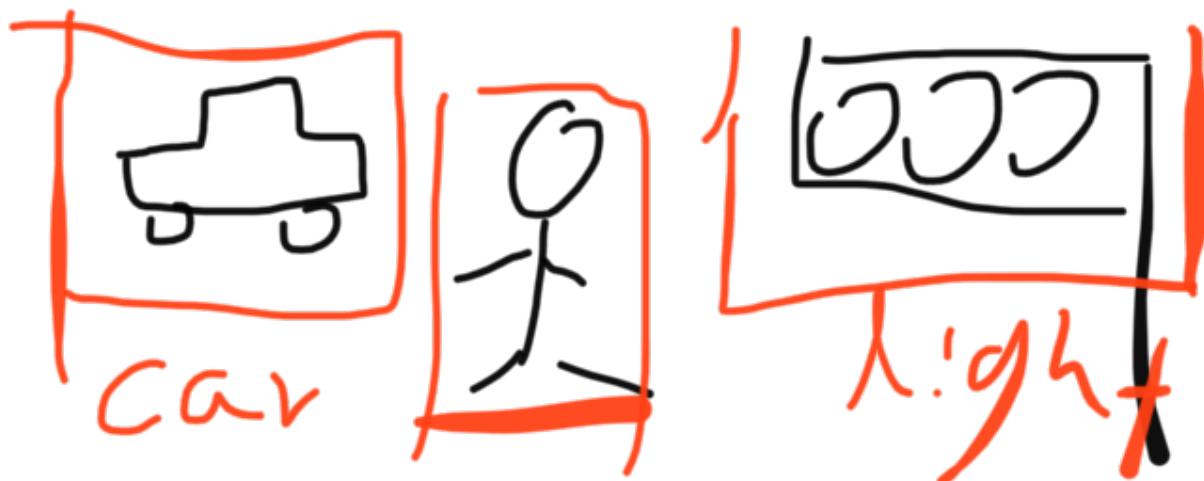
When multi-task learning makes sense

- 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 $\frac{1,000,000}{1,000}$
 - B $\frac{1,000}{1,000}$
- Can train a big enough neural network to do well on all the tasks.

▶ 🔍 9:47 / 12:59 Andrew Ng

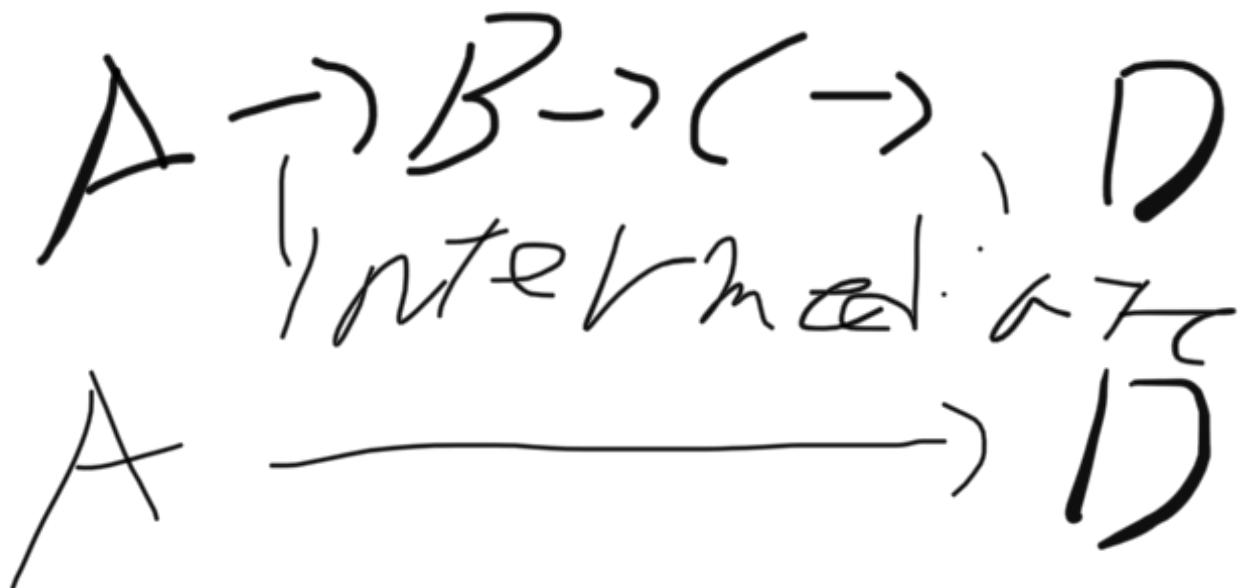
Save Note Discuss Download

Like Comment Print



- Not common as transfer learning
- Since it's difficult to find or set up so many different tasks you would want to train a single neural network.

End to End deep learning



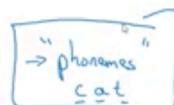
End to End approach directly aim result

Whether to use end-to-end deep learning

Pros and cons of end-to-end deep learning

Pros:

- Let the data speak $x \rightarrow y$
- Less hand-designing of components needed



Cons:

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

Data Hand-design