Error Analysis

Carrying out error analysis
Look at dev examples to evaluate ideas

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.

90% accuracy
\rightarrow 10\% error

\Rightarrow 5-10\ min

\Rightarrow 5/100
\Rightarrow 9.5\%
\Rightarrow 50/100
\Rightarrow 50.1\%
\Rightarrow 100\%
\Rightarrow 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

<table>
<thead>
<tr>
<th>Image</th>
<th>Dog</th>
<th>Great Cats</th>
<th>Blurry</th>
<th>Instagram</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>✓</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>Pitbull</td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>Raining day at 200</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>% of total</td>
<td>8%</td>
<td>43%</td>
<td>61%</td>
<td>12%</td>
<td></td>
</tr>
</tbody>
</table>
Error Analysis

Cleaning up Incorrectly labeled data
Incorporately labeled examples

\[ \begin{array}{cccccc}
\text{x} & \text{dog} & \text{cat} & \text{dog} & \text{cat} & \text{dog} \\
\text{y} & 1 & 0 & 1 & 1 & 0 & 1 \\
\end{array} \]

Training set

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

Systematic errors

Andrew Ng
## Error analysis

<table>
<thead>
<tr>
<th>Image</th>
<th>Dog</th>
<th>Great Cat</th>
<th>Blurry</th>
<th>Incorrectly labeled</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>98</td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>Labeler missed cat in background</td>
</tr>
<tr>
<td>99</td>
<td>✔️</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td>✔️</td>
<td>Drawing of a cat; Not a real cat.</td>
</tr>
</tbody>
</table>

| % of total | 8%  | 43%  | 61%  | 6%  |

Overall dev set error \[10\%\]

Errors due incorrect labels \[0.6\%\]

Errors due to other causes \[9.4\%\]

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.

• Train and dev/test data may now come from slightly different distributions.
Error Analysis

Build your first system quickly, then iterate
Speech recognition example

- Noisy background
  - Café noise
  - Car noise
- Accented speech
- Far from microphone
- 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.
Mismatched training and dev/test data

Training and testing on different distributions
Cat app example

Data from webpages

Data from mobile app

Option 1:

Option 2:

Andrew Ng
Speech recognition example

Training

- Purchased data
- Smart speaker control
- Voice keyboard

Dev/test

- Speech activated rearview mirror

...
Mismatched training and dev/test data

Bias and Variance with mismatched data distributions
Cat classifier example

Assume humans get ≈ 0% error.

Training error .... 10% ↓ 9%
Dev error .... 10%

Training-dev set: Same distribution as training set, but not used for training
Bias/variance on mismatched training and dev/test sets

Human level

Training set error
Training - dev set error
\rightarrow Dev error
\rightarrow Test error

\begin{align*}
4\% & \uparrow \text{ avoidable bias} \\
7\% & \uparrow \text{ variance} \\
10\% & \uparrow \text{ data mismatch} \\
12\% & \uparrow \text{ degree of ambiguity} \\
12\% & \rightarrow \text{ dev set.}
\end{align*}

\begin{align*}
4\% \\
7\% \ \} \\
10\% \ \} \\
6\% \ \} \\
6\%
\end{align*}
More general formulation

- Human level
  - General speech recognition
  - "Human level" 4% → 6%
  - "Training error" 7% → 6%
  - "Training - dev error" 10% → "Dev/Test error" 6%

- Error on examples trained on
- Error on examples not trained on

- Data mismatch
- Variance
- Avoidable bias
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

Overtfit to 1 hour of car noise

10,000 hours

Set of all audio in car
Artificial data synthesis

Car recognition:

20 cars

synthesized

All cars
Learning from multiple tasks

Transfer learning
Transfer learning

\[ x \rightarrow \text{pre-training} \rightarrow \text{fine-tuning} \rightarrow \hat{y} \]

\[ x \rightarrow \text{audio} \rightarrow \text{speech recognition} \rightarrow \hat{y} \]

\[ x \rightarrow \text{image recognition} \rightarrow \text{radiology diagnosis} \]

\[ x \rightarrow \text{image recognition} \rightarrow 1,000,000 \rightarrow 100 \rightarrow \hat{y} \]

\[ x \rightarrow \text{speech recognition} \rightarrow 10,000h \rightarrow \text{wakeword/trigger detection} \rightarrow 50h \]

Andrew Ng
When transfer learning makes sense

\[ \text{Task from A} \rightarrow \text{B} \]

• 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.
Learning from multiple tasks

Multi-task learning
Simplified autonomous driving example

![Image of a car and a stop sign]

- Pedestrians
- Cars
- Stop signs
- Traffic lights

\[ y = \begin{bmatrix} y_1^{(i)} & y_2^{(i)} & y_3^{(i)} & \ldots & y_m^{(i)} \end{bmatrix} \]

Andrew Ng
Neural network architecture

Loss: \[ \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{4} y_{i}^{(j)} \log \hat{y}_{i}^{(j)} - y_{i}^{(j)} \log (1 - \hat{y}_{i}^{(j)}) \]

Unlike softmax regression:
- One input can have multiple labels
- Sum only over \( \hat{y}_{i}^{(j)} \) with 0/1 label

\[ X \xrightarrow{\hat{y}} \]

Multi-task learning: \[ Y = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \]
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.
- Can train a big enough neural network to do well on all the tasks.
End-to-end deep learning

What is end-to-end deep learning
What is end-to-end learning?

Speech recognition example

```
3,000h
↑
```

```
10,000h
↓
```

```
100,000h
```

“cat”

Audio → MFCC → Features → Phonemes → Words → Transcript

Audio → Audio → ... → Transcript

Andrew Ng
Face recognition

Have data for each of 2 sub-tasks.
More examples

Machine translation

\[ (x, y) \quad \Rightarrow \quad \text{English} \quad \Rightarrow \text{text analysis} \quad \Rightarrow \quad \text{French} \]

Estimating child’s age:

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

Image \xrightarrow{} \text{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
• Less hand-designing of components needed

Cons:
• May need large amount of data
• Excludes potentially useful hand-designed components
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 $y$?