These slides are distributed under the Creative Commons License.

DeepLearning.AI makes these slides available for educational purposes. You may not use or distribute these slides for commercial purposes. You may make copies of these slides and use or distribute them for educational purposes as long as you cite DeepLearning.AI as the source of the slides.

For the rest of the details of the license, see https://creativecommons.org/licenses/by-sa/2.0/legalcode
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 \( \approx 100 \) mislabeled dev set examples.
- Count up how many are dogs.

\[
\begin{align*}
5/100 &= 5\% \\
95/100 &= 95\%
\end{align*}
\]
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
Incorrectly labeled examples

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
</tr>
</thead>
<tbody>
<tr>
<td>🐶</td>
<td>1</td>
</tr>
<tr>
<td>🐱</td>
<td>0</td>
</tr>
<tr>
<td>🐱</td>
<td>1</td>
</tr>
<tr>
<td>🐱</td>
<td>1</td>
</tr>
<tr>
<td>🐶</td>
<td>0</td>
</tr>
<tr>
<td>🐕</td>
<td>1</td>
</tr>
<tr>
<td>🐱</td>
<td>1</td>
</tr>
</tbody>
</table>

Training set.

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

Systematic errors
## 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>
<tr>
<td>% of total</td>
<td>8%</td>
<td>43%</td>
<td>61%</td>
<td>6%</td>
<td></td>
</tr>
</tbody>
</table>

Overall dev set error: 100%
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 $\approx 0\%$ error.

Training error: $1\%$ (9/10)
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
→ Dev error
→ Test error

4% ↑ avoidable bias
7% ↑ variance
10% ↑ data mismatch
12% ↑ degree of difficulty
12% → dev set.

4%  
7%  
10%  
6%  
6%
More general formulation

- General speech recognition
- Reuse mirror

- "Human level" 4%
- "Training error" 7%
- "Training - Dev error" 10%
- "Dev/Test error" 6%

- Human level
- Error on examples trained on
- Error on examples not trained on

- Avoidable bias
- Variance

- Data mismatch
Mismatches in 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

Synthesize

Set of all audio in car

Outfit to 1 hour of car noise

Andrew Ng
Artificial data synthesis

Car recognition:
Learning from multiple tasks

Transfer learning
Transfer learning
When transfer learning makes sense

- 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

deeplearning.ai
Simplified autonomous driving example

- Pedestrians
- Cars
- Stop signs
- Traffic lights

\[ y = \begin{bmatrix} y^{(1)} & y^{(2)} & y^{(3)} & \ldots & y^{(m)} \end{bmatrix} \]
Neural network architecture

Loss: $\ell \left( \frac{y_i}{1+y_i} \right) = \frac{1}{m} \sum_{i=1}^{m} \sum_{j=1}^{n} \ell \left( \hat{y}_{ij}, y_{ij} \right)$

- $\sum$ over all $i$ with a label $y_{ij} = 1$.
- Unlike softmax regression, one image can have multiple labels.

$\ell \left( \hat{y}_{ij}, y_{ij} \right)$:

- Usual logistic loss:
  \[-y_{ij} \log \hat{y}_{ij} - (1 - y_{ij}) \log (1 - \hat{y}_{ij})\]

Multi-task learning:

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

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.

• 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

```
input audio → MFCC → features → phonemes → words → transcript
```

- 3,000h
- 10,000h
- 100,000h

\[ \text{"cat"} \]
Face recognition

[Image courtesy of Baidu]

Have data for each of 2 subtasks.

Andrew Ng
More examples

Machine translation

\[(x, y) \rightarrow \text{English} \rightarrow \text{text analysis} \rightarrow \cdots \rightarrow \text{French}\]

Estimating child’s age:

1. Image \(\rightarrow\) bones \(\rightarrow\) age
2. Image \(\rightarrow\) 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 $y$?

\[ x \rightarrow y \]

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