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
Introduction to ML strategy

Why ML Strategy?
Motivating example

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
  - ...

Andrew Ng
Introduction to ML strategy

Orthogonalization
TV tuning example

Orthogonalization

\[ 0.1 \times \]
\[ + 0.5 \times \]
\[ - 1.7 \times \]
\[ + 0.8 \times \]
\[ + \ldots \]

\[ \rightarrow 0.3 \times \text{angle} - 0.8 \times \text{speed} \]
\[ \rightarrow 2 \times \text{angle} + 0.9 \times \text{speed} \]

Car

\[ \rightarrow \text{Steering} \]
\[ \rightarrow \{ \text{Accelerate, Braking} \} \]

Andrew Ng
Chain of assumptions in ML

→ Fit training set well on cost function
  - (≈ human-level performance)

→ Fit dev set well on cost function

→ Fit test set well on cost function

→ Performs well in real world
  - (Happy cat pic app users)
Setting up your goal

Single number evaluation metric
Using a single number evaluation metric

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>95%</td>
<td>90%</td>
</tr>
<tr>
<td>B</td>
<td>98%</td>
<td>85%</td>
</tr>
</tbody>
</table>

\[ F_1 \text{ score} = \frac{2 \times P \times R}{P + R} \]

Dev set + Single number evaluation metric
red: speed up iterating
## Another example

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>US</th>
<th>China</th>
<th>India</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>3%</td>
<td>7%</td>
<td>5%</td>
<td>9%</td>
</tr>
<tr>
<td>B</td>
<td>5%</td>
<td>6%</td>
<td>5%</td>
<td>10%</td>
</tr>
<tr>
<td>C</td>
<td>2%</td>
<td>3%</td>
<td>4%</td>
<td>5%</td>
</tr>
<tr>
<td>D</td>
<td>5%</td>
<td>8%</td>
<td>7%</td>
<td>2%</td>
</tr>
<tr>
<td>E</td>
<td>4%</td>
<td>5%</td>
<td>2%</td>
<td>4%</td>
</tr>
<tr>
<td>F</td>
<td>7%</td>
<td>11%</td>
<td>8%</td>
<td>12%</td>
</tr>
</tbody>
</table>
Setting up your goal

Satisficing and optimizing metrics
Another cat classification example

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Accuracy</th>
<th>Running time</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>90%</td>
<td>80ms</td>
</tr>
<tr>
<td>B</td>
<td>92%</td>
<td>95ms</td>
</tr>
<tr>
<td>C</td>
<td>95%</td>
<td>1,500ms</td>
</tr>
</tbody>
</table>

Cost = \(\text{Accuracy} - 0.5 \times \text{Running Time}\)

Maximize Accuracy
Subject to Running Time ≤ 100 ms.

\(N\) metric: \(1\) optimizing
\(N-1\) satisfying
Setting up your goal

Train/dev/test distributions
Cat classification dev/test sets

development set, held out cross validation corp

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

Randomly shuffle into dev/test
True story (details changed)

Optimizing on dev set on loan approvals for medium income zip codes

\[ \uparrow \quad x \rightarrow y \text{ (repay loan?)} \]

Tested on low income zip codes

\( \sim 3 \text{ month} \)
Guideline

Choose a dev set and test set to reflect data you expect to get in the future and consider important to do well on.
Setting up your goal

Size of dev and test sets
Old way of splitting data

- 70/1 for training
- 30/1 for test
- 60/1 for training
- 30/1 for dev
- 20/1 for test
- 1,000,000
- 10,000
- 100
- 1000
- 10,000
Size of dev set

Set your dev set to be big enough to detect differences in algorithm/models you’re trying out.

- 100: small  
  - 1%
- 1,000
- 10,000
- 100,000

97% → 97.1%
0.1%

Only advertise

0.01%
0.001%

Andrew Ng
Set your test set to be big enough to give high confidence in the overall performance of your system.
When to change dev/test sets and metrics

Setting up your goal
Cat dataset examples

- Metric: classification error
  - Algorithm A: 3% error
  - Algorithm B: 5% error

\[
\text{Error} = \frac{1}{m_{\text{dev}}} \sum_{i=1}^{m_{\text{dev}}} \left( \mathbb{1}\{y_{\text{pred}} + y^{(i)}\} \right) \\
\rightarrow w^{(i)} = \begin{cases} 
1 & \text{if } x^{(i)} \text{ is non-porn} \\
10 & \text{if } x^{(i)} \text{ is porn}
\end{cases}
\]
Orthogonalization for cat pictures: anti-porn

→ 1. So far we’ve only discussed how to define a metric to evaluate classifiers.

→ 2. Worry separately about how to do well on this metric.

$$ J = \frac{1}{\sum w_i} \sum_{i=1}^{m} w_i L (\hat{y}(i), y(i)) $$
Another example

Algorithm A: 3% error

Algorithm B: 5% error

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.
Comparing to human-level performance

Why human-level performance?
Comparing to human-level performance

Accuracy vs. Time

- X → Y
- Audio → Transcript
- Image → cat(0/1)
- Bayes error
- Bayes optimal error
- Best possible error
- Human

Andrew Ng
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.
Comparing to human-level performance

Avoidable bias
Bias and Variance

- **High bias** (underfitting)
- "Just right"
- **High variance** (overfitting)
Bias and Variance

Cat classification

Training set error: __________

Dev set error: __________

high variance  high bias  high bias  low bias

Human-level 20%
Cat classification example

Humans (\% Bayes) 8\%

Training error 8\%

Dev error 10\%

Avoidable bias 0.5\%

Variance 2.7\%

Focus on bias

Focus on variance

Human-level error as a proxy for Bayes error.
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

What is “human-level” error?
Error analysis example

Human (proxy for Bayes error)

Training error

Dev error

\[
\begin{align*}
&\text{Bias} \quad \text{Variance} \\
&0.7\% \quad 0.2\% \\
&0.5\% \quad 0.0\% \\
&0.7\% \\
&0.8\%
\end{align*}
\]
Summary of bias/variance with human-level performance

Human-level error
(proxy for Bayes error)

Training error

Dev error

Available bias

Variance
Comparing to human-level performance

Surpassing human-level performance
Surpassing human-level performance

Team of humans: 0.5%  
One human: 0.1%  
Training error: 0.6%  
Dev error: 0.8%  

What is available 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, ...
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.

2. The training set performance generalizes pretty well to the dev/test set.
Reducing (avoidable) bias and variance

Human-level

Training error

 avoidable bias

Dev error

Train bigger model

Train longer/better optimization algorithms
- Momentum, RMSprop, Adam

NN architecture/hyperparameters search

More data

Regularization
- $L_2$, dropout, data augmentation

NN architecture/hyperparameters search

Andrew Ng