Introduction to ML strategy

Why ML Strategy?

deeplearning.ai
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
  - ...
Introduction to ML strategy

Orthogonalization

deeplearning.ai
TV tuning example

Orthogonalization

Car

\[ 0.1 \times \]

\[ + 0.3 \times \]

\[ - 1.7 \times \]

\[ + 0.8 \times \]

\[ + \ldots \]

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

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

\[ \rightarrow \text{Steering} \]

\[ \rightarrow \{ \text{Accelera} \}

\[ \rightarrow \{ \text{Braking} \}

Andrew Ng
Chain of assumptions in ML

→ Fit training set well on cost function
→ 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

Idea

Experiment

Code

<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>

\[ F_1 \text{ score } = \frac{2 \cdot \frac{1}{P} + \frac{1}{R}}{\frac{1}{P} + \frac{1}{R}} \text{, "Harmonic mean"} \]

Dev set + Single number evaluation metric

real 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 \( \text{Accuracy} \)
Subject to \( \text{Running Time} \leq 100 \text{ ms} \)

N matrix: 1 optimizing
N-1 satisfying

Watson's / Trigger words
- Alexa, OK Google
- Hey Siri, Baidu

Accuracy

#false positive

Maximize accuracy
s.t. \( \leq 1 \) false positive every 24 hours.

Andrew Ng
Setting up your goal

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

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

Idea
- development set, held out cross validation set

Experiment
- Randomly shuffle into dev/test

Code

Andrew Ng
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

- 100
- 100
- 10,000

- 70.1%
  - Train
- 30.1%
  - Test

- 60.1%
  - Train
- 20.1%
  - Dev
- 20.1%
  - Test

- 98.1%
  - Train
  - 1,000,000
  - 10,500

Andrew Ng
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%
Set your test set to be big enough to give high confidence in the overall performance of your system.

- Train + Dev.
- 10,000
- 150,000

No test set.
Setting up your goal

When to change dev/test sets and metrics
Cat dataset examples

→ Metric: classification error

Algorithm A: 3% error

√ Algorithm B: 5% error

\[
\text{Error} = \frac{1}{\sum_{i=1}^{m_{\text{train}}}} \sum_{i=1}^{m_{\text{train}}} \left( \mathbb{1}\{y_{\text{pred}} + y(i)\} \right) \left( x(i) \right)
\]

→ \( w(i) = \begin{cases} 
1 & \text{if } x(i) \text{ is non-porn} \\
10 & \text{if } x(i) \text{ is porn}
\end{cases} \)
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.

\[ J = \frac{1}{m} \sum_{i=1}^{m} \sum_{j} w_j \ell \big( \hat{y}_j^{(i)}, y_j^{(i)} \big) \]
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

- $X \rightarrow Y$
- Audio $\rightarrow$ Transcript
- Image $\rightarrow$ cat (0/1)

Accuracy vs. Time

Bayes error

Best possible error

Human
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”**: Good fit
- **High variance**: Overfitting
Bias and Variance

Cat classification

Human-level 20% ...

Training set error: ___

Dev set error:

high variance high bias high bias low bias

low variance
Cat classification example

Humans (bias)

Training error 8%
Dev error 10%

Focus on bias

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?

\[ \text{Bayes error} \leq 0.5 \% \]
Error analysis example

Human (proxy for Bayes error)

Training error

Dev error

$\text{Human (proxy for Bayes error)}$

$\text{Training error}$

$\text{Dev error}$
Summary of bias/variance with human-level performance

- Human-level error
  (proxy for Bayes error)
- Training error
- Dev error
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?

Andrew Ng
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
    - EEG, Skin care,...
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

Dev error

Avoidable bias

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