MLOps: From Model-centric to Data-centric AI

Andrew Ng
AI system = Code + Data
(model/algorithm)
Inspecting steel sheets for defects

Baseline system: 76.2% accuracy
Target: 90.0% accuracy
Audience poll: Should the team improve the code or the data?

Poll results:
### Improving the code vs. the data

<table>
<thead>
<tr>
<th></th>
<th>Steel defect detection</th>
<th>Solar panel</th>
<th>Surface inspection</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Baseline</strong></td>
<td>76.2%</td>
<td>75.68%</td>
<td>85.05%</td>
</tr>
<tr>
<td><strong>Model-centric</strong></td>
<td>+0%</td>
<td>+0.04%</td>
<td>+0.00%</td>
</tr>
<tr>
<td></td>
<td>(76.2%)</td>
<td>(75.72%)</td>
<td>(85.05%)</td>
</tr>
<tr>
<td><strong>Data-centric</strong></td>
<td>+16.9%</td>
<td>+3.06%</td>
<td>+0.4%</td>
</tr>
<tr>
<td></td>
<td>(93.1%)</td>
<td>(78.74%)</td>
<td>(85.45%)</td>
</tr>
</tbody>
</table>

Andrew Ng
Data is Food for AI

~1% of AI research?

80%

~99% of AI research?

20%

PREP
Source and prepare high quality ingredients
Source and prepare high quality data

ACTION
Cook a meal
Train a model

Andrew Ng
Lifecycle of an ML Project

1. Define project
2. Collect data
3. Train model
4. Deploy in production

- Define project
- Define and collect data
- Training, error analysis & iterative improvement
- Deploy, monitor and maintain system

Andrew Ng
Decide to work on speech recognition for voice search
Collect Data: Speech Recognition

Scope project

Collect data

Train model

Deploy in production

Define and collect data

Is the data labeled consistently?

“Um, today’s weather”
“Um... today’s weather”
“Today’s weather”
Iguana Detection Example

Labeling instruction:

Use bounding boxes to indicate the position of iguanas
Making data quality systematic: MLOps

• Ask two independent labelers to label a sample of images.

• Measure consistency between labelers to discover where they disagree.

• For classes where the labelers disagree, revise the labeling instructions until they become consistent.
Labeler consistency example

Steel defect detection (39 classes). Class 23: Foreign particle defect.
Labeler consistency example

Steel defect detection (39 classes). Class 23: Foreign particle defect.
## Making it systematic: MLOps

<table>
<thead>
<tr>
<th><strong>Model-centric view</strong></th>
<th><strong>Data-centric view</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Collect what data you can, and develop a model good enough to deal with the noise in the data.</td>
<td>The consistency of the data is paramount. Use tools to improve the data quality; this will allow multiple models to do well.</td>
</tr>
<tr>
<td>Hold the data fixed and iteratively improve the code/model.</td>
<td><strong>Hold the code fixed and iteratively improve the data.</strong></td>
</tr>
</tbody>
</table>
Audience poll: Think about the last supervised learning model you trained. How many training examples did you have? Please enter an integer.

Poll results:
Kaggle Dataset Size

No. of Training Examples in Kaggle Datasets

- 0 - 1K: 4
- 1K - 10K: 8
- 10K - 100K: 4
- 100K+: 4

Andrew Ng
Small Data and Label Consistency

- Small data
- Noisy labels

- Big data
- Noisy labels

- Small data
- Clean (consistent) labels

Andrew Ng
Theory: Clean vs. noisy data

You have 500 examples, and 12% of the examples are noisy (incorrectly or inconsistently labeled).

The following are about equally effective:
• Clean up the noise
• Collect another 500 new examples (double the training set)

With a data centric view, there is significant room for improvement in problems with <10,000 examples!
Example: Clean vs. noisy data

Accuracy (mAP)

Clean

Noisy

Number of training examples

250 500 750 1000 1250 1500

0.3 0.4 0.5 0.6 0.7 0.8

Note: Big data problems where there's a long tail of rare events in the input (web search, self-driving cars, recommender systems) are also small data problems.

Andrew Ng
Train model: Speech Recognition

Error analysis shows your algorithm does poorly in speech with car noise in the background. What do you do?

Model-centric view
How can I tune the model architecture to improve performance?

Data-centric view
How can I modify my data (new examples, data augmentation, labeling, etc.) to improve performance?
Train model: Speech Recognition

Making it systematic – iteratively improving the data:
• Train a model
• Error analysis to identify the types of data the algorithm does poorly on (e.g., speech with car noise)
• Either get more of that data via data augmentation, data generation or data collection (change inputs $x$) or give more consistent definition for labels if they were found to be ambiguous (change labels $y$)
Deploy: Speech Recognition

Monitor performance in deployment, and flow new data back for continuous refinement of model.

- Systematically check for concept drift/data drift (performance degradation)
- Flow data back to retrain/update model regularly

Andrew Ng
Making it systematic: The rise of MLOps

Andrew Ng
Making it systematic: The rise of MLOps

$\text{AI systems} = \text{Code} + \text{Data}$

- Creation: Software engineers
- Quality/Infrastructure: DevOps
- MLOps: ML engineers

Andrew Ng
Traditional software vs AI software

Traditional software

Scope project → Develop code → Deploy in production

AI software

Scope project → Collect data → Train model → Deploy in production

Andrew Ng
MLOps: Ensuring consistently high-quality data

- Scope project
- Collect data
- Train model
- Deploy in production

How do I define and collect my data?
How do I modify data to improve model performance?
What data do I need to track concept/data drift?
Audience poll: Who do you think is best qualified to take on an MLOps role?

Poll results:

Who do you think is best qualified to take on an MLOps role?

- Machine Learning engineer / Data scientist: 56.6%
- DevOps engineer: 13.7%
- Software engineer: 6.7%
- Domain expert (e.g., manufacturing expert, linguist/speech expert, etc.): 23%
MLOps’ most important task: Ensure consistently high-quality data in all phases of the ML project lifecycle.

Good data is:
• Defined consistently (definition of labels $y$ is unambiguous)
• Cover of important cases (good coverage of inputs $x$)
• Has timely feedback from production data (distribution covers data drift and concept drift)
• Sized appropriately
Takeaways: Data-centric AI

MLOps’ most important task is to make high quality data available through all stages of the ML project lifecycle.

AI system = Code + Data

**Model-centric AI**
How can you change the model (code) to improve performance?

**Data-centric AI**
How can you systematically change your data (inputs x or labels y) to improve performance?

Important frontier: MLOps tools to make data-centric AI an efficient and systematic process.