De-risking machine learning projects using novel technologies

Daniel Situnayake, Founding Engineer
Acquire valuable data securely

Enrich data & train ML algorithms

Validate ML with real-time data

Build optimized embedded inference

10,000+ Enterprises

20,000+ Developers

44,000+ Projects

49M+ Datasets
“Gartner predicts that through 2022, 85 percent of AI projects will deliver erroneous outcomes due to bias in data, algorithms or the teams responsible for managing them.”
Unique constraints of embedded ML

- Resource-constrained devices
- Lack of relevant datasets
- Limited feedback loop once deployed
Problem:
Device constraints limit the size of our models.
Model compression can amplify bias

- Conventional model architectures too large & inefficient for embedded
- Compression impacts performance on long tail
- May result in poor real-world performance

Figure 1: Most natural image datasets exhibit a long-tail distribution with an unequal frequency of attributes in the training data. Below each attribute sub-group in CelebA, we report the share of training set and total frequency count.

Characterizing the bias of compressed models, S Hooker et al.
Philosophy:

Don’t just rely on compression. Create embedded-first architectures.
SSD object detection not suited to MCUs

Figure 6. Mobile-Det: SSD-based detection with MobileNet as backbone, modified based on [14]
SSD object detection not suited to MCUs

https://pjreddie.com/darknet/yolov1
Take advantage of use case constraints
Object detection on MCU

- Fast for custom models
- Works on MCU class hardware
- Really fast on embedded Linux
- Detect smaller and more objects better
Problem:

Datasets are not available for common embedded use cases.
Keyword spotting needs a lot of data

~560 samples, class balanced

### Last training performance (validation set)

<table>
<thead>
<tr>
<th></th>
<th>ACCURACY</th>
<th>LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>%</td>
<td>75.2%</td>
<td>1.11</td>
</tr>
</tbody>
</table>

### Confusion matrix (validation set)

<table>
<thead>
<tr>
<th></th>
<th>NO</th>
<th>NOISE</th>
<th>UNKNOWN</th>
<th>YES</th>
</tr>
</thead>
<tbody>
<tr>
<td>NO</td>
<td>66.2%</td>
<td>0%</td>
<td>18.2%</td>
<td>13.6%</td>
</tr>
<tr>
<td>NOISE</td>
<td>3.7%</td>
<td>85.2%</td>
<td>11.1%</td>
<td>0%</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>15.6%</td>
<td>9.4%</td>
<td>56.3%</td>
<td>18.8%</td>
</tr>
<tr>
<td>YES</td>
<td>3.1%</td>
<td>6.3%</td>
<td>0%</td>
<td>90.6%</td>
</tr>
<tr>
<td>F1 SCORE</td>
<td>0.68</td>
<td>0.84</td>
<td>0.63</td>
<td>0.83</td>
</tr>
</tbody>
</table>
Keyword spotting needs a lot of data

~10.8k samples, class balanced

19x the data for 12% better accuracy

### Last training performance (validation set)

<table>
<thead>
<tr>
<th></th>
<th>ACCURACY</th>
<th>LOSS</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>%</strong></td>
<td>87.3%</td>
<td>0.41</td>
</tr>
</tbody>
</table>

### Confusion matrix (validation set)

<table>
<thead>
<tr>
<th></th>
<th>NOISE</th>
<th>OFF</th>
<th>ON</th>
<th>UNKNOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOISE</td>
<td>95.4%</td>
<td>0.2%</td>
<td>0.4%</td>
<td>4%</td>
</tr>
<tr>
<td>OFF</td>
<td>2.0%</td>
<td>87.1%</td>
<td>5.6%</td>
<td>5.4%</td>
</tr>
<tr>
<td>ON</td>
<td>4.2%</td>
<td>6.1%</td>
<td>86.4%</td>
<td>3.3%</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>10.3%</td>
<td>4.9%</td>
<td>4.5%</td>
<td>80.3%</td>
</tr>
<tr>
<td><strong>F1 SCORE</strong></td>
<td>0.90</td>
<td>0.88</td>
<td>0.88</td>
<td>0.83</td>
</tr>
</tbody>
</table>
Philosophy:

Reduced data requirements means more successful projects.
Few-shot transfer learning for keyword spotting

Few-Shot Keyword Spotting in Any Language

- As few as 5-10 samples
- Multilingual (even unseen languages and words)
- Higher accuracy than with numerous samples
Intelligent dataset augmentation

- More data is always helpful
- Not all data contributes equally
- Smart use of data increases performance and reduces training time
Intelligent dataset augmentation - how?

- Use representation learning to create an embedding model
- Identify “hard negatives” from supplementary data library
- Build training curriculum based on predicted difficulty
Problem:

Test dataset performance does not represent real world performance.
What does this mean?

Model testing results

<table>
<thead>
<tr>
<th></th>
<th>HELLO WORLD</th>
<th>NOISE</th>
<th>UNKNOWN</th>
<th>UNCERTAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>HELLO WORLD</td>
<td>91.0%</td>
<td>0.8%</td>
<td>1.6%</td>
<td>6.6%</td>
</tr>
<tr>
<td>NOISE</td>
<td>0.7%</td>
<td>86.7%</td>
<td>7.0%</td>
<td>5.6%</td>
</tr>
<tr>
<td>UNKNOWN</td>
<td>3.4%</td>
<td>2.7%</td>
<td>88.5%</td>
<td>5.4%</td>
</tr>
<tr>
<td>F1 SCORE</td>
<td>0.93</td>
<td>0.91</td>
<td>0.90</td>
<td></td>
</tr>
</tbody>
</table>
What does this mean?

“Was every 1 second sample in the test dataset classified correctly?”
Philosophy:

Try to predict real-world performance as early as possible.
Realistic application testing

- Test on realistic samples: 24 hours of real world audio
- Measure & visualize industry standard metrics (FAR, FRR)
- Understand impact of post-processing
Realistic application testing

- Use genetic algorithm to design optimal post-processing configuration

- Choose your ideal balance between false accepts and false rejects

- Account for device constraints and latency
Conclusions

- Embedded ML involves more risk than conventional ML
- Risk can be minimized using novel approaches
- Be creative and try new things!
Oura Ring Gen 3
Edge Impulse Lucky Draw
Thank you!

p.s. we’re hiring :)

hello@edgeimpulse.com

Edge Impulse
3031 Tisch Way  110 Plaza West
San Jose, CA 95128 USA