Machine Learning as a Service: 
The Challenges of Serving a Million 
Client Distributions

Sunita Sarawagi
CSE and CMInDS at IITB

Joint work with: Vihari Piratla, Soumen Chakrabarti, Preethi Jyothi, Abhijeet Awasthi, Kartik K., Praneeth Netrapalli

PAKDD 12th May 2021
State of the art Machine Learning Models

- Record-breaking accuracy on many difficult tasks in vision, speech, and NLP.
- Training such models is highly resource-intensive
  - Size of labelled data: E.g. Automatic Speech Recognition (ASR) 100,000 hours!
  - Computing resources: Hundreds of GPUs/TPUs
  - Size of the models: GPT-3 has 175B parameters
- Affordable only to a few resource rich organizations
Machine Learning Services

Several clients

$D_1 \hspace{1cm} D_2 \hspace{1cm} D_3 \hspace{1cm} \cdots \hspace{1cm} \cdots \hspace{1cm} \cdots \hspace{1cm} D_{1000000}$

$x \downarrow \hspace{1cm} y, \text{conf} \uparrow$

Machine Learning Cloud APIs

Server Model

Huge amount of (proprietary) training data
Cloud Machine Learning APIs

● Computer vision
  ○ Amazon Rekognition, Azure Custom Vision
  ○ Service, Google Cloud Vision API

● Speech recognition
  ○ Amazon Transcribe, Azure Custom Speech Service, Google Dialogflow Enterprise Edition

● Natural language processing
  ○ Amazon Comprehend, Azure Web Language Model API, Google Cloud Natural Language API, IBM Watson
Machine Learning Services

Several clients

- $D_1$: Coffee shop in UK
- $D_2$: Retail store in India
- $D_3$: Help center in AUS
- ... $D_{1000000}$

Huge amount of (proprietary) training data

- $x$: Input
- $y$, $\text{conf}$: Output

Machine Learning Cloud APIs

- $\theta$: Model

Human effort

$\text{CO}_2$
## Cloud-ML Vs Classical ML

<table>
<thead>
<tr>
<th></th>
<th>Classical-ML</th>
<th>Cloud-ML</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training</strong></td>
<td>I.I.D samples from a distribution D, need to generalize to samples from the same distribution D</td>
<td>Samples from mixture of several distributions $D_1, D_2, ..., D_k$, need to generalize to these and new distributions $D_{k+1}, ...$</td>
</tr>
<tr>
<td><strong>Adaptation</strong></td>
<td></td>
<td><strong>Domain generalization</strong></td>
</tr>
<tr>
<td><strong>Evaluation</strong></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
# Cloud-ML Vs Classical ML

<table>
<thead>
<tr>
<th></th>
<th>Classical-ML</th>
<th>Cloud-ML</th>
</tr>
</thead>
</table>
| **Training**     | I.I.D samples from a distribution $D$, need to generalize to samples from the same distribution $D$ | Samples from mixture of several distributions $D_1, D_2, \ldots, D_k$, need to generalize to these and new distributions $D_{k+1}, \ldots$  
Domain generalization |
| **Adaptation**   | Fine-tune to adapt parameters to new setup.                                  | Model hidden, client-specific adaptation expensive at server.  
Black-box adaptation |
<p>| <strong>Evaluation</strong>   |                                                                                |                                                                                           |</p>
<table>
<thead>
<tr>
<th></th>
<th>Classical-ML</th>
<th>Cloud-ML</th>
</tr>
</thead>
</table>
| **Training**             | I.I.D samples from a distribution $D$, need to generalize to samples from the same distribution $D$ | Samples from mixture of several distributions $D_1, D_2, \ldots, D_k$, need to generalize to these and new distributions $D_{k+1}, \ldots$  
                                      | Domain generalization                                                        |                                                                                                                                 |
| **Adaptation**           | Fine-tune to adapt parameters to new setup.                                  | Model hidden, client-specific adaptation expensive at server.  
                                      | Black-box adaptation                                                        |                                                                                                                                 |
| **Evaluation**           | Single number (BLEU, WER, F1) on a benchmark dataset                        | Single accuracy cannot suffice. Accuracy is a surface over client data characteristics.        |
Machine Learning Services

Several clients

\[ D_1, D_2, \ldots, D_k, D_{k+1}, \ldots, D_{1000000} \]

Server Model

Multi-domain labeled data over subset of domains

Huge amount of training data

No client/domain id

Machine Learning Cloud APIs

\[ L = \{(x, y, d)\} \]

\[ d \in D_1, D_2, \ldots, D_k \]
Domain Generalization Problem
Automatic Speech Recognition

Train

Test
Training for Domain Generalization

OCR model trained with 20 fonts but new fonts during test
Digit recognition with rotation as domain

Domain Invariant features:
- Number of edges: 3
- Number of corners: 3
- Angle between 1, 2, or 3

Domain Specific Features:
- Angle of 1 = 90 or 90±15.
- Angle of 2 = 45 or 45±15.
- Angle of 3 = 0 or 0±15.
Approach 1: Domain Erasure

Domain adversarial networks (DANs): train $h$ so that adversary cannot find any domain signal (wipe out domain signals)

1. NN could overfit on training domains
2. Accuracy of in-domain data suffers when domain and label not separable
3. Unstable training

Approach 2: Data Augmentation

- When label and domain are entangled,
- Augment with examples from hallucinated domains.
- CrossGradient training:
  1. Use a domain classifier to extract latent continuous features of domain
  2. Augment training data with perturbed training instances along directions of domain change

Performance

Limitations

1. Domain classifier may not use all domain-revealing feature.
2. Perturbations are conservative.

Approach 3: Common Specific Decomposition

Factorization of softmax parameters: $W_d = W_c + \gamma_d W_s$

Add orthogonality constraint $W_c \perp W_s$ to make solution identifiable

During training: loss on $W_c$ and loss on $W_d$

Only use $W_c$ during test time!

Speech Tasks

- CSD is consistently better.
- Decreasing gains as number of train domains increase.

Improvement over baseline on speech task for varying number of domains, shown on X-axis.
Handwritten character recognition tasks

- LRD, CG, MASF are strong contemporary baselines.
- CSD consistently outperforms others.

Accuracy gains over the ERM baseline.
Domain Generalization: Summary

- **CSD:** A simple and efficient solution to DG
  - Runtime of only x1.1 of ERM baseline
  - All other approaches significantly slower to train.
  - Practical and easy to implement: Only modify final layer.
  - May not be great for in-domain data: hybrid of CSD & CrossGrad?

- **Topic of much active research**
  - Fairness --- worst case group accuracy
  - Heterogeneity of domains during training.
### Cloud-ML Vs Classical ML

<table>
<thead>
<tr>
<th></th>
<th>Classical-ML</th>
<th>Cloud-ML</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Training</strong></td>
<td>I.I.D samples from a distribution $D$, need to generalize to samples from the same distribution $D$</td>
<td>Samples from mixture of several distributions $D_1, D_2, ..., D_k$, need to generalize to these and new distributions $D_{k+1}, ...$ Domain generalization</td>
</tr>
<tr>
<td><strong>Adaptation</strong></td>
<td>Fine-tune to adapt parameters to new setup.</td>
<td>Model hidden, client-specific adaptation expensive at server. Black-box adaptation</td>
</tr>
<tr>
<td><strong>Evaluation</strong></td>
<td>Single number (BLEU, WER, F1) on a benchmark dataset</td>
<td>Single accuracy cannot suffice. Accuracy is a surface over client data characteristics.</td>
</tr>
</tbody>
</table>
Domain Adaptation

- Given a trained model $\Theta$ on a large dataset $D$
- A **small** amount of labelled data $L$ in a target domain $Q$
  - $L$ is not large enough to train a good model on its own
  - Want to get the best of $\Theta$ and $L$
- **Blackbox adaptation:**
  - Want to do this without access and knowledge of $D$
  - Cannot update $\Theta$
Domain adaptation approaches

- Fine-tuning $\Theta$ with gradient descent on small $L$
  1. Initialize with $\Theta$
  2. MLE ($L$) + Regularize to be close to $\Theta$

- Domain adversarial methods:
  - Represent server’s data $D$ and client’s so they look similar

- Meta-Learning: exploit multi-domain data to learn to adapt
  - MAML: requires nested loops during training
  - SNAIL: self-attention over per-domain data

They all require updating and access to service parameters!
Black-Box Adaptation of ASR for Accented Speech

Kartik Khandelwal, Preethi Jyothi, Abhijeet Awasthi, Sunita Sarawagi
Indian Institute of Technology Bombay, Mumbai, India
Introduction

- Client has limited labeled data of target accent
- Service not trained on that accent but overall better.
- Service: proprietary, black-box
  - Cannot fine-tune to adapt to the target accent

Goal: Adapt a black-box ASR system to speech from a target accent
Machine Learning Services

Server Model

Trained on $O(50,000)$ hours

En-US ASR Model

Service ASR

22% WER

Client

Indian accent

$D_1 \sim 37$ hours

Open Source Models (DS2)

100 hours

Local ASR

Finetune

28% WER

Can we perform better than both?
Possible solution: Alignment

Align and combine the two word sequences (ROVER)

\[ c, q \]

\[ s, p \]

Gold everyone toasted the
everyone posted the
0.9 0.4 0.6 1.0

Merged everyone to posted the
0.9 0.6 1.0

Accent errors might affect only a part of the word.

\[ c \]: Transcribed Text
\[ q \]: word-level confidence
\[ s \]: Transcribed Text
\[ p \]: word-level confidence
Service Vs Local Model

- Local ASR model expected to be good only on accented parts of an audio
- Service model better in all other portions
  - More noise-tolerant
  - Stronger language model
- Need to merge at finer level? Character-level?
  - Service does not provide character-level confidences.
  - everyone toostdd the
Our Approach: FineMerge

Exploit white-box access to local model

Frame-level character distributions: $Q = Q_1, \ldots, Q_T$

Intervene with service transcript here.
FineMerge Algorithm: Workflow

**Local ASR**

**Service ASR**

**Q**

**S**

**Q_s**

**Beam decode**

**Final transcript**

---

**Table:**

<table>
<thead>
<tr>
<th>frame t</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>d_t, Q_t(d_t)</td>
<td>t 0.99</td>
<td>t 0.99</td>
<td>o 1.0</td>
<td>- 0.98</td>
<td>s 0.93</td>
<td>t 0.99</td>
<td>a 0.55</td>
<td>t 0.64</td>
</tr>
<tr>
<td>2</td>
<td>S_t, Q_t(S_t)</td>
<td>- 6e-5</td>
<td>p 0.0</td>
<td>o 1.0</td>
<td>- 0.01</td>
<td>s 0.93</td>
<td>t 0.99</td>
<td>e 0.44</td>
<td>d 0.29</td>
</tr>
<tr>
<td>3</td>
<td>r_t, Q_t^s(r_t)</td>
<td>t 0.62</td>
<td>t 0.99</td>
<td>o 1.0</td>
<td>- 0.98</td>
<td>s 0.93</td>
<td>t 0.99</td>
<td>e 0.66</td>
<td>d 0.57</td>
</tr>
</tbody>
</table>

Where:

- \( d_t = \text{argmax}_c Q_t(c) \)
- \( S_t = \text{aligned service char at } t \)
- \( r_t = \text{argmax}_c P_t^s(c) \)
## WER with different service models

<table>
<thead>
<tr>
<th>Method</th>
<th>Indian</th>
<th>Australian</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>G-US</td>
<td>G-Video</td>
</tr>
<tr>
<td>Local</td>
<td></td>
<td>28.0</td>
</tr>
<tr>
<td>Service</td>
<td>22.3</td>
<td>13.8</td>
</tr>
<tr>
<td>Rover</td>
<td>21.1</td>
<td>20.5</td>
</tr>
<tr>
<td>LM rescore</td>
<td>22.1</td>
<td>13.4</td>
</tr>
<tr>
<td>FineMerge</td>
<td>18.4</td>
<td>13.4</td>
</tr>
</tbody>
</table>

**FineMerge** gives lowest WER and achieves improvements of up to 28% in WER over the service model.
<table>
<thead>
<tr>
<th>Gold Service Local Rover FineMerge</th>
<th>Indian</th>
<th>Australian</th>
</tr>
</thead>
<tbody>
<tr>
<td>everyone toasted the ..</td>
<td>nora finds herself ugly ..</td>
<td></td>
</tr>
<tr>
<td>everyone posted the ..</td>
<td>nora van to self ugly ..</td>
<td></td>
</tr>
<tr>
<td>everyone to state the ..</td>
<td>nor iphones herself ugly ..</td>
<td></td>
</tr>
<tr>
<td>everyone to posted the ..</td>
<td>nor to self ugly ..</td>
<td></td>
</tr>
<tr>
<td>everyone toasted the ..</td>
<td>nora finds herself ugly ..</td>
<td></td>
</tr>
<tr>
<td>for a brief time ..</td>
<td>hannelore is an ..</td>
<td></td>
</tr>
<tr>
<td>soda beef time ..</td>
<td>i don’t know what is an ..</td>
<td></td>
</tr>
<tr>
<td>for a breezen time ..</td>
<td>hailar is an ..</td>
<td></td>
</tr>
<tr>
<td>for a beef time ..</td>
<td>i don’t know what is an ..</td>
<td></td>
</tr>
<tr>
<td>for a <strong>brief</strong> time ..</td>
<td><strong>hannelore</strong> is an ..</td>
<td></td>
</tr>
<tr>
<td>the condition also occurs ..</td>
<td>..rope a bull while on a</td>
<td></td>
</tr>
<tr>
<td>definition of circus ..</td>
<td>..work a bowl while on a</td>
<td></td>
</tr>
<tr>
<td>the condition also acres ..</td>
<td>..rope the ball while on a</td>
<td></td>
</tr>
<tr>
<td>the definition also circus ..</td>
<td>..work a bowl while on a</td>
<td></td>
</tr>
<tr>
<td>the condition also <strong>occurs</strong></td>
<td>..rope a <strong>bull</strong> while on a</td>
<td></td>
</tr>
</tbody>
</table>
Summary

- FineMerge couples an open-source accent-tuned local model with the black-box service.
- Achieves up to 28% reduction in WER over service APIs of varying grades of quality.
- Helps recover words that were present in neither the Black-Box service and local ASR model.
Future work

- Blackbox adaptation of translation models? Image recognition models?
- On-the-fly adaptation with unlabelled data

Only scratched the surface, lot of fresh-thinking needed to redesign training, adaptation, and evaluation pipelines so a model can keep a million clients happy.
Thank you!