Frontiers and Open-Challenges

CS330
Logistics

Final project presentations next week
Schedule on Piazza.

Final project report
Due next Friday midnight.

This is the last lecture!
We’ll leave time for course evaluations at the end.
Today: What doesn’t work very well?
(and how might we fix it)

Meta-learning for addressing distribution shift
Capturing equivariances with meta-learning
Adapting to distribution shift

What does it take to run multi-task & meta-RL across distinct tasks?
what set of distinct tasks do we train on?
what challenges arise?

Open Challenges
Why address distribution shift?
Our current paradigm (ML research)

dataset → model → evaluation

Our current reality

stocks

robots

Can our algorithms handle the changing world?
How does industry cope?

Chip Huyen on misperceptions about ML production:

3. If nothing happens, model performance remains the same
   
   ML models perform best right after training. In prod, ML systems degrade quickly bc of concept drift.

Tip: train models on data generated 6 months ago & test on current data to see how much worse they get.

4. You won’t need to update your models as much
   
   One mindboggling fact about DevOps: Etsy deploys 50 times/day. Netflix 1000s times/day. AWS every 11.7 seconds.

MLOps isn’t an exemption. For online ML systems, you want to update them as fast as humanly possible.

the way our techniques are being *used* != the way we *intend*
Can we discover **equivariant** and **invariant structure** via meta-learning? (i.e. symmetries)

One solution to distribution shift: build in structure to solve this problem. 
- e.g. convolutions

+ Great when we know the structure & how to build it in!  
- Not great when we don’t
Does MAML already do this?

MAML can learn **equivariant** initial features but equivariance **may not be preserved** in the gradient update!

Goal: Can we decompose weights into **equivariant structure** & **corresponding parameters**?

If so: update **only parameters** in the inner loop, retaining **equivariance**.
How are equivariances represented in neural networks?

Let’s look at an example.

1D convolution layer

1D convolution represented as FC layer

Representing Equivariance by Reparametrization

Key idea: reparametrize weight matrix $W$

$W \cdot x = y$

1D convolution represented as FC layer

Theoretically, this can directly represent *decoupled* equivariant sharing pattern + filter parameters. for all $G$-convolutions with finite group $G$
Meta-Learning Equivariance

**Inner loop**: only update parameters $v \rightarrow v'$, keep equivariance $U$ fixed

**Outer loop**: learn equivariance $U$ and initial parameters $v$

Important assumption: Some symmetries shared by all tasks.
Can we recover convolutions?

from translationally equivariant data

Mean-squared error on held-out test tasks

<table>
<thead>
<tr>
<th>Method</th>
<th>$k = 1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML-FC</td>
<td>$3.2 \pm .29$</td>
</tr>
<tr>
<td>MAML-LC</td>
<td>$2.4 \pm .23$</td>
</tr>
<tr>
<td>MAML-Conv</td>
<td>$0.16 \pm .02$</td>
</tr>
<tr>
<td>MSR-FC (Ours)</td>
<td>$0.18 \pm .03$</td>
</tr>
</tbody>
</table>

MAML-X: X corresponds to architecture
(fully-connected, locally-connected, convolution)

MSR-FC: fully-connected layer weights $W$

Can we recover something better than convolutions?

…from data with partial translation symmetry

<table>
<thead>
<tr>
<th>Method</th>
<th>$k = 1$</th>
<th>$k = 2$</th>
<th>$k = 5$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML-FC</td>
<td>3.2 ± .29</td>
<td>2.1 ± .15</td>
<td>.89 ± .05</td>
</tr>
<tr>
<td>MAML-LC</td>
<td>2.4 ± .23</td>
<td>1.6 ± .11</td>
<td>.81 ± .05</td>
</tr>
<tr>
<td>MAML-Conv</td>
<td>.16 ± .02</td>
<td>.52 ± .05</td>
<td>.44 ± .02</td>
</tr>
<tr>
<td>MSR-FC (Ours)</td>
<td>.18 ± .03</td>
<td>.21 ± .02</td>
<td>.22 ± .01</td>
</tr>
</tbody>
</table>

$k$: rank of a locally-connected layer

…from data with translation + rotation + reflection symmetry

<table>
<thead>
<tr>
<th>Rotation/Flip Equivariance MSE</th>
<th>Rot</th>
<th>Rot+Flip</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td></td>
<td></td>
</tr>
<tr>
<td>MSR-Conv (Ours)</td>
<td>.004</td>
<td>.001</td>
</tr>
<tr>
<td>MAML-Conv</td>
<td>.504</td>
<td>.507</td>
</tr>
</tbody>
</table>

MSR-Conv: $W$ corresponds to convolution layer weights

Can we learn symmetries from augmented data?

Algorithm 2: Augmentation Meta-Training

<table>
<thead>
<tr>
<th>Method</th>
<th>5 way</th>
<th>Aug-Omniglot</th>
<th>20 way</th>
<th>Aug-MiniImagenet</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1-shot</td>
<td>5-shot</td>
<td>1-shot</td>
<td>5-shot</td>
</tr>
<tr>
<td>MAML</td>
<td>87.3 ± 0.5</td>
<td>93.6 ± 0.3</td>
<td>67.0 ± 0.4</td>
<td>79.9 ± 0.3</td>
</tr>
<tr>
<td>MAML (Big)</td>
<td>89.3 ± 0.4</td>
<td>94.8 ± 0.3</td>
<td>69.6 ± 0.4</td>
<td>83.2 ± 0.3</td>
</tr>
<tr>
<td>ANIL</td>
<td>86.4 ± 0.5</td>
<td>93.2 ± 0.3</td>
<td>67.5 ± 3.5</td>
<td>79.8 ± 0.3</td>
</tr>
<tr>
<td>ProtoNets</td>
<td>92.9 ± 0.4</td>
<td>97.4 ± 0.2</td>
<td>85.1 ± 0.3</td>
<td>94.3 ± 0.2</td>
</tr>
<tr>
<td>MSR (Ours)</td>
<td>95.3 ± 0.3</td>
<td>97.7 ± 0.2</td>
<td>84.3 ± 0.2</td>
<td>92.6 ± 0.2</td>
</tr>
</tbody>
</table>

MSR provides a framework for understanding the interplay of features & structure in meta-learning.

Today: What doesn’t work very well?
(and how might we fix it)

Meta-learning for addressing distribution shift
Capturing equivariances with meta-learning
Adapting to distribution shift

What does it take to run multi-task & meta-RL across distinct tasks?
what set of distinct tasks do we train on?
what challenges arise?

Open Challenges
What kind of distribution shift to adapt to?
We’ll now focus on: **group shift**

categorical group variable $z$
e.g. user, location, time of day(can be derived from meta-data)

Training data from $p(x, y | z)p_{tr}(z)$
Test data from $p(x, y | z)p_{ts}(z)$
can capture label shift, most covariate shift
captures problems like federated learning

**Group DRO** (*distributionally robust optimization*):
(Ben-Tal et al. ’13, Duchi et al ’16)

Form adversarial distribution $q(z)$: $\min_{\theta} \sup_{q \in \mathcal{Q}} \mathbb{E}_{q z} \left[ \mathbb{E}_{p_{x y | z}} \left[ \ell(g(x; \theta), y) \right] \right]$

+ can enable robust solutions  - often sacrifices average/empirical group performance
+ less pessimistic than adversarial robustness
Can we aim to *adapt* instead of aiming for robustness?

**Test time**

unlabeled data from test sub-distribution  
(e.g. new user, different time-of-day, new place)  

adapt model & infer labels

Assumption: test inputs from one group available in a batch or streaming.

Adaptive risk minimization (ARM)

Adaptive risk minimization (ARM)

1. Construct sub-distributions of training data
2. Train for adaptation to sub-distributions.

How to adapt with unlabeled data?

MAML with learned loss

\[ \theta' = \theta - \alpha \nabla_{\theta} \mathcal{L}(\theta, x_{1:3}) \]

or

meta-learning with context variable

Simplest setting: context = BN statistics

---

Experimental Comparisons

ERM - standard deep network training

DRNN - distributional robustness
(Sagawa, Koh et al. ICLR ’20)

UW - ERM but upweight groups to the uniform distribution

ARM - adaptive risk minimization
  ARM-CML - adapt with context variable
  ARM-BN - adapt using batch norm stats
  ARM-LL - adapt with learned loss
Experiment 1. Federated Extended MNIST (Cohen et al. 2017, Caldas et al. 2019)

**Distribution shift**: adapt to new users with only unlabeled data

<table>
<thead>
<tr>
<th>Method</th>
<th>WC</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>ERM</td>
<td>62.9 ± 1.9</td>
<td>80.1 ± 0.9</td>
</tr>
<tr>
<td>UW*</td>
<td>61.8 ± 0.9</td>
<td>80.1 ± 0.3</td>
</tr>
<tr>
<td>DRNN</td>
<td>58.1 ± 0.7</td>
<td>74.4 ± 0.8</td>
</tr>
<tr>
<td>q-FedAvg [37]</td>
<td>58.2 ± 1.0</td>
<td>80.8 ± 0.3</td>
</tr>
<tr>
<td>ARM-CML</td>
<td>67.8 ± 1.3</td>
<td>85.7 ± 0.3</td>
</tr>
<tr>
<td>ARM-BN</td>
<td>72.6 ± 0.3</td>
<td>85.7 ± 0.1</td>
</tr>
<tr>
<td>ARM-LL</td>
<td>69.6 ± 2.1</td>
<td>85.6 ± 0.5</td>
</tr>
</tbody>
</table>

+ 5% **improvement** in average accuracy
+ 10% **improvement** in worst-case accuracy

**ARM** - adaptive risk minimization  
**ERM** - standard deep network training  
**UW** - ERM but upweight groups to the uniform distribution  
**q-FedAvg** (Li et al. 2020) - federated learning method

Experiment 1. Federated Extended MNIST (Cohen et al. 2017, Caldas et al. 2019)

Distribution shift: adapt to new users with only unlabeled data
**Experiment 2. CIFAR-C, TinyImageNet-C** (Hendrycks & Dietterich, 2019)

**Distribution shift:** adapt to *new* image corruptions
(train using 56 corruptions, test using 22 disjoint corruptions)

<table>
<thead>
<tr>
<th>Method</th>
<th>CIFAR-10-C</th>
<th>Tiny ImageNet-C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>WC</td>
<td>Avg</td>
</tr>
<tr>
<td>ERM</td>
<td>49.6 ± 0.1</td>
<td>69.8 ± 0.4</td>
</tr>
<tr>
<td>UW*</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>DRNN</td>
<td>44.5 ± 0.5</td>
<td>70.7 ± 0.6</td>
</tr>
<tr>
<td>ARM-CML</td>
<td>67.7 ± 0.5</td>
<td>79.2 ± 0.3</td>
</tr>
<tr>
<td>ARM-BN</td>
<td>71.1 ± 0.1</td>
<td>80.9 ± 0.2</td>
</tr>
<tr>
<td>ARM-LL</td>
<td>66.9 ± 0.2</td>
<td>75.7 ± 0.3</td>
</tr>
</tbody>
</table>


ARM - adaptive risk minimization

DRNN - distributional robustness

(Stagawa, Koh et al. ICLR '20)

ERM - standard deep network training

UW - ERM but upweight groups to the uniform distribution

+ 3-10% improvement in average accuracy

+ 8-21% improvement in worst-case accuracy
Today: What doesn’t work very well? (and how might we fix it)

Meta-learning for addressing distribution shift
Capturing equivariances with meta-learning
Adapting to distribution shift

Takeaways

Preliminary evidence that meta-learning can capture *equivariances* via reparametrized weight matrices

*Allow adaptation / fine-tuning without* labeled target data via adaptive risk minimization
Today: What doesn’t work very well?
(and how might we fix it)

Meta-learning for addressing distribution shift
Capturing equivariances with meta-learning
Adapting to distribution shift

What does it take to run multi-task & meta-RL across distinct tasks?
what set of distinct tasks do we train on?
what challenges arise?

Open Challenges
Have MAML, RL², PEARL, DREAM accomplished our goal of making policy adaptation fast?
Sort of…

Can we adapt to *entirely new tasks*?

\[
\text{meta-train task distribution} = \text{meta-test task distribution}
\]

\[\land \text{not sparse}\]

\[\Rightarrow \text{Need broad distribution of tasks for meta-training}\]

A few options:

Our desiderata

50+ qualitatively distinct tasks
shaped reward function & success metrics
All tasks individually solvable (to allow us to focus on multi-task / meta-RL component)
Unified state & action space, environment (to facilitate transfer)

Meta-World Benchmark

T Yu, D Quillen, Z He, R Julian, K Hausman, C Finn, S Levine. Meta-World. CoRL ’19
Results: Meta-learning algorithms seem to struggle…

<table>
<thead>
<tr>
<th>Methods</th>
<th>ML45 meta-train</th>
<th>ML45 meta-test</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RL²</td>
<td></td>
<td></td>
</tr>
<tr>
<td>PEARL</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

…even on the 45 meta-training tasks!

Multi-task RL algorithms also struggle…

<table>
<thead>
<tr>
<th>Methods</th>
<th>MT50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multi-task PPO</td>
<td>8.98%</td>
</tr>
<tr>
<td>Multi-task TRPO</td>
<td>22.86%</td>
</tr>
<tr>
<td>Task embeddings</td>
<td>15.31%</td>
</tr>
<tr>
<td>Multi-task SAC</td>
<td>28.83%</td>
</tr>
<tr>
<td>Multi-task multi-head SAC</td>
<td><strong>35.85%</strong></td>
</tr>
</tbody>
</table>

T Yu, D Quillen, Z He, R Julian, K Hausman, C Finn, S Levine. *Meta-World*. CoRL ‘19
Why the poor results?

Exploration challenge?  All tasks individually solvable.

Data scarcity?  All methods given budget with plenty of samples.

Limited model capacity?  All methods plenty of capacity.

Training models *independently* performs the best.

Our conclusion: must be a multi-task *optimization* challenge.
Hypothesis 1: Gradients from different tasks often conflict

If so: would see negative inner product of gradients.

Hypothesis 2: When they do conflict, they cause more damage than expected.

i.e. due to high curvature

Our solution: try to avoid making other tasks worse, when taking gradient step.

Algorithm:

If two gradients conflict:
project each onto the normal plane of the other

Else:
leave them alone

i.e. project conflicting gradients
“PCGrad”

Multi-Task RL on Meta-World:

MT10

MT50

Success Rates vs. Number of thousand env steps for different methods:

- SAC+PA
- Multi-head SAC+PA
- Independent
- SAC+PCGrad+PA (ours)

also helps multi-task supervised learning, complementary to multi-task architectures.
Why does it work?

(Part 1)

Why does it work? (Part 2)

Hypothesis 1: Gradients from different tasks often conflict

If so: would see negative inner product of gradients

Hypothesis 2: When they do conflict, they cause more damage than expected.

i.e. due to high curvature & difference in grad magnitude

1. conflicting gradients
2. large positive curvature
3. difference in gradient magnitude

"tragic triad"

Is PCGrad provably better under these three conditions?

Are these three conditions actually why we see improvements on large-scale problems?

Why does it work?
(Part 2)

1. conflicting gradients
2. large positive curvature
3. difference in gradient magnitude

Is PCGrad provably better under these three conditions?

short answer: yes, if large enough conflict, curvature, gradient magnitude difference

(long answer)

Theorem 2. Suppose $\mathcal{L}$ is differentiable and the gradient of $\mathcal{L}$ is Lipschitz continuous with constant $L > 0$. Let $\theta^{MT}$ and $\theta^{PCGrad}$ be the parameters after applying one update to $\theta$ with $g$ and PCGrad-modified gradient $g^{PC}$ respectively, with step size $t > 0$. Moreover, assume $H(\mathcal{L}; \theta, \theta^{MT}) \geq \ell \|g\|^2_2$ for some constant $\ell \leq L$, i.e. the multi-task curvature is lower-bounded. Then $\mathcal{L}(\theta^{PCGrad}) \leq \mathcal{L}(\theta^{MT})$ if

(a) $\cos \phi_{12} \leq -\Phi(g_1, g_2)$,
(b) $\ell \geq \xi(g_1, g_2)L$, and
(c) $t \geq \frac{\ell}{\ell - \xi(g_1, g_2)L}$.

Proof. See Appendix B.

Are these three conditions actually why we see improvements on large-scale problems?

Today: What doesn’t work very well? (and how might we fix it)

What does it take to run multi-task & meta-RL across distinct tasks?

what set of distinct tasks do we train on?
what challenges arise?

**Takeaways**

Scaling to broad task distributions is hard, can’t be taken for granted:

- Train on broad, dense task distributions
- Avoid conflicting gradients
Today: What doesn’t work very well?
(and how might we fix it)

Meta-learning for addressing distribution shift
Capturing equivariances with meta-learning
Adapting to distribution shift

What does it take to run multi-task & meta-RL across distinct tasks?
what set of distinct tasks do we train on?
what challenges arise?

Open Challenges
Open Challenges in Multi-Task and Meta Learning

(that we haven't previously covered)
Open Challenges in Multi-Task and Meta Learning

Addressing fundamental problem assumptions

- Generalization: Out-of-distribution tasks, long-tailed task distributions
The problem with long-tailed distributions.

We learned how to do few-shot learning

...but these few-shot tasks may be from a different distribution

We've seen some generalization to the tail:
- prototypical clustering networks for dermatological diseases
- adaptive risk minimization

Further hints might come from domain adaptation, robustness literature.
Open Challenges in Multi-Task and Meta Learning

Addressing fundamental problem assumptions

- Generalization: Out-of-distribution tasks, long-tailed task distributions
- Multimodality: Can you learn priors from multiple modalities of data?
Rich sources of prior experiences.

Can we learn priors across multiple data modalities?
Varying dimensionalities, units
Carry different, complementary forms of information

Some hints might come from multimodal learning literature.

Open Challenges in Multi-Task and Meta Learning

Addressing fundamental problem assumptions
- Generalization: Out-of-distribution tasks, long-tailed task distributions
- Multimodality: Can you learn priors from multiple modalities of data?
- Algorithm, Model Selection: When will multi-task learning help you?

Benchmarks
- Breadth: That challenge current algorithms to find common structure
- Realistic: That reflect real-world problems
Some steps towards good benchmarks

**Meta-Dataset**
Triantafillou et al. ‘19

**Meta-World Benchmark**
Yu et al. ‘19

**Visual Task Adaptation Benchmark**
Zhai et al. ‘19

**Taskonomy Dataset**
Zamir et al. ‘18

**Goal:** reflection of real world problems + appropriate level of difficulty + ease of use
Open Challenges in Multi-Task and Meta Learning

Addressing fundamental problem assumptions

- Generalization: Out-of-distribution tasks, long-tailed task distributions
- Multimodality: Can you learn priors from multiple modalities of data?
- Algorithm, Model Selection: When will multi-task learning help you?

Benchmarks

- Breadth: That challenge current algorithms to find common structure
- Realistic: That reflect real-world problems

Improving core algorithms

- Computation & Memory: Making large-scale bi-level optimization practical
- Theory: Develop a theoretical understanding of the performance of these algorithms
- Multi-Step Problems: Performing tasks in sequence presents challenges.

+ the challenges you discovered in your homework & final projects!
The Bigger Picture
Machines are *specialists*.
Humans are generalists.

Source: https://youtu.be/8vNxjwt2AqY
A Step Towards Generalists

Some of what we covered in CS330:

- learn multiple tasks (multi-task learning)
- leverage prior experience when learning new things (meta-learning)
- learn general-purpose models (model-based RL)
- prepare for tasks before you know what they are (exploration, skill discovery, unsupervised meta-learning)
- perform tasks in sequence (hierarchical RL)
- learn continuously (lifelong learning)

What’s missing?
Reminders

Final project presentations next week
  Schedule on Piazza.

Final project report
  Due next Friday midnight.

This is the last lecture!
  We’ll leave time for course evaluations at the end.