The Peculiar Optimization and Regularization Challenges in Multi-Task Learning and Meta-Learning

Chelsea Finn
By Braque or Cezanne?
How did you accomplish this?

Through previous experience.
How might you get a machine to accomplish this task?

Modeling image formation
  Geometry
SIFT features, HOG features + SVM
Fine-tuning from ImageNet features
Domain adaptation from other painters

Can we explicitly learn priors from previous experience that lead to efficient downstream learning?

Can we learn to learn?
Outline

1. Brief overview of meta-learning

2. A peculiar yet ubiquitous problem in meta-learning
   (and how we might regularize it away)

3. Can we scale meta-learning to broad task distributions?
How does meta-learning work? An example.

Given 1 example of 5 classes:

- Training data \( D_{\text{train}} \)

Classify new examples:

- Test set \( X_{\text{test}} \)
How does meta-learning work? An example.

Given 1 example of 5 classes:

Classify new examples

training data $D_{\text{train}}$  

test set $X_{\text{test}}$
How does meta-learning work?

One approach: parameterize learner by neural network

\[ y_{ts} = f(\mathcal{D}_{tr}, x_{ts}; \theta) \]

(Hochreiter et al. '91, Santoro et al. '16, many others)
How does meta-learning work?

Another approach: embed optimization inside the learning process

\[ y_{ts} = f(\mathcal{D}_{tr}, x_{ts}; \theta) \]

(Maclaurin et al. ’15, Finn et al. ’17, many others)
Can we learn a representation under which RL is fast and efficient?

Can we learn a representation under which imitation is fast and efficient?

subset of training objects

input demo
(via teleoperation)

resulting policy

[real-time execution]

The Bayesian perspective

$\theta$ \rightarrow \phi_j \rightarrow x_{j_n} \rightarrow N \rightarrow J$

meta-learning $\sim$ learning priors $p(\phi | \theta)$ from data

(Grant et al. ’18, Gordon et al. ’18, many others)
Outline

1. Brief overview of meta-learning

2. A peculiar yet ubiquitous problem in meta-learning
   (and how we might regularize it away)

3. Can we scale meta-learning to broad task distributions?
How we construct tasks for meta-learning.

Randomly assign class labels to image classes for each task $\mathcal{T}_1$, $\mathcal{T}_2$, $\mathcal{T}_3$.

- $\mathcal{D}_{tr}$: Training data
- $x_{ts}$: Testing data

Tasks are mutually exclusive.

Algorithms must use training data to infer label ordering.
What if label order is consistent?

Tasks are **non-mutually exclusive**: a single function can solve all tasks.

The network can simply learn to classify inputs, irrespective of $\mathcal{D}_{tr}$.
The network can simply learn to classify inputs, irrespective of $\mathcal{D}_{tr}$.
What if label order is consistent?

For new image classes: can't make predictions w/o $\mathcal{D}_{tr}$

<table>
<thead>
<tr>
<th></th>
<th>NME Omniglot</th>
<th>20-way 1-shot</th>
<th>20-way 5-shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML</td>
<td>7.8 (0.2)%</td>
<td>50.7 (22.9)%</td>
<td></td>
</tr>
</tbody>
</table>
Is this a problem?

- **No**: for image classification, we can just shuffle labels*
- **No**, if we see the same image classes as training (& don’t need to adapt at meta-test time)
- But, **yes**, if we want to be able to adapt with data for new tasks.
Another example

If you tell the robot the task goal, the robot can ignore the trials.

T Yu, D Quillen, Z He, R Julian, K Hausman, C Finn, S Levine. Meta-World. CoRL ’19
Another example

Model can memorize the canonical orientations of the training objects.

Yin, Tucker, Yuan, Levine, Finn. Meta-Learning without Memorization. ICLR’19
Can we do something about it?
If tasks **mutually exclusive**: single function cannot solve all tasks
(i.e. due to label shuffling, hiding information)

If tasks are **non-mutually exclusive**: single function can solve all tasks

*multiple solutions* to the meta-learning problem

\[ y^{ts} = f_\theta(D_i^{tr}, x^{ts}) \]

**One solution:** memorize canonical pose info in \( \theta \) & ignore \( D_i^{tr} \)

**Another solution:** carry no info about canonical pose in \( \theta \), acquire from \( D_i^{tr} \)

An entire *spectrum of solutions* based on how *information* flows.

Suggests a potential approach: control information flow.

If tasks are *non-mutually exclusive*: single function can solve all tasks

*multiple solutions* to the meta-learning problem

\[ y^{ts} = f_\theta(D_{i}^{tr}, x^{ts}) \]

**One solution:** memorize canonical pose info in \( \theta \) & ignore \( D_i^{tr} \)

**Another solution:** carry no info about canonical pose in \( \theta \), acquire from \( D_i^{tr} \)

An entire *spectrum of solutions* based on how *information* flows.

---

**Meta-regularization**

one option: \( \max I(\hat{y}_{ts}, D_{tr} \mid x_{ts}) \)

minimize meta-training loss + information in \( \theta \)

\[ \mathcal{L}(\theta, D_{meta-train}) + \beta D_{KL}(q(\theta; \theta_{\mu}, \theta_{\sigma}) \| p(\theta)) \]

Places precedence on using information from \( D_{tr} \) over storing info in \( \theta \).

Can combine with your favorite meta-learning algorithm.

Yin, Tucker, Yuan, Levine, Finn. *Meta-Learning without Memorization*. ICLR’19
**Omniglot** without label shuffling: “non-mutually-exclusive” Omniglot

<table>
<thead>
<tr>
<th>Method</th>
<th>NME Omniglot</th>
<th>20-way 1-shot %</th>
<th>20-way 5-shot %</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAML</td>
<td>7.8 (0.2)</td>
<td>50.7 (22.9)</td>
<td></td>
</tr>
<tr>
<td>TAML</td>
<td>9.6 (2.3)</td>
<td>67.9 (2.3)</td>
<td></td>
</tr>
<tr>
<td>MR-MAML (W) (ours)</td>
<td>83.3 (0.8)</td>
<td>94.1 (0.1)</td>
<td></td>
</tr>
</tbody>
</table>

On **pose prediction** task:

<table>
<thead>
<tr>
<th>Method</th>
<th>MAML</th>
<th>MR-MAML (W) (ours)</th>
<th>CNP</th>
<th>MR-CNP (W) (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSE</td>
<td>5.39 (1.31)</td>
<td><strong>2.26 (0.09)</strong></td>
<td>8.48 (0.12)</td>
<td>2.89 (0.18)</td>
</tr>
</tbody>
</table>

(and it’s not just as simple as standard regularization)

<table>
<thead>
<tr>
<th>CNP</th>
<th>CNP + Weight Decay</th>
<th>CNP + BbB</th>
<th>MR-CNP (W) (ours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>8.48 (0.12)</td>
<td>6.86 (0.27)</td>
<td>7.73 (0.82)</td>
<td><strong>2.89 (0.18)</strong></td>
</tr>
</tbody>
</table>


Does meta-regularization lead to better generalization?

Let $P(\theta)$ be an arbitrary distribution over $\theta$ that doesn’t depend on the meta-training data. (e.g. $P(\theta) = \mathcal{N}(\theta; 0, I)$)

For MAML, with probability at least $1 - \delta$,

$$\text{er}(\theta_\mu, \theta_\sigma) \leq \frac{1}{n} \sum_{i=1}^{n} \hat{\text{er}}(\theta_\mu, \theta_\sigma, D_i, D_i^*) + \left( \sqrt{\frac{1}{2(K-1)}} + \sqrt{\frac{1}{2(n-1)}} \right) \sqrt{D_{KL}(\mathcal{N}(\theta; \theta_\mu, \theta_\sigma) \| P) + \log \frac{n(K+1)}{\delta}},$$

generalization error error on the meta-training set meta-regularization

With a Taylor expansion of the RHS + a particular value of $\beta$ $\rightarrow$ recover the MR MAML objective.

Proof: draws heavily on Amit & Meier ‘18
2. A peculiar yet ubiquitous problem in meta-learning
(and how we might regularize it away)

*Intermediate Takeaways*

**meta overfitting**
memorize training functions $f_i$
corresponding to tasks in your meta-training dataset

**standard overfitting**
memorize training datapoints $(x_i, y_i)$
in your training dataset

**meta regularization**
controls information flow
regularizes description length
of meta-parameters

**standard regularization**
regularize hypothesis class
(though not always for DNNs)

Outline

1. Brief overview of meta-learning

2. A peculiar yet ubiquitous problem in meta-learning
   (and how we might regularize it away)

3. Can we scale meta-learning to broad task distributions?
Has meta-learning accomplished our goal of making adaptation fast?

Sort of…

Can adapt to:

- new objects
- new goal velocities
- new object categories

Can we adapt to entirely new tasks or datasets?
Can we adapt to entirely *new* tasks or datasets?

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

\[\rightarrow\] Need **broad** distribution of tasks for meta-training

Can we look to RL benchmarks?

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</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>meta-train</td>
<td>meta-test</td>
</tr>
<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 an *optimization* challenge.
Prior literature on multi-task learning

Architectural solutions:

- Multi-head architectures
- FiLM: Visual Reasoning with a General Conditioning Layer. Perez et al. ‘17
- Sluice Networks. Ruder, Bingel, Augenstein, Sogaard ‘17
- Deep Relation Networks. Long, Wang ’15
- Cross-Stitch Networks. Misra, Shrivastava, Gupta, Hebert ‘16
- Multi-Task Attention Network. Liu, Johns, Davison ‘18

Task weighting solutions:

\[ L_{\text{tot}} = w_{\text{depth}} L_{\text{depth}} + w_{\text{kpt}} L_{\text{kpt}} + w_{\text{normals}} L_{\text{normals}} \]

GradNorm. Chen et al. ‘18

\[ \min_{\theta^1, \ldots, \theta^T} \sum_{t=1}^{T} \epsilon^t \hat{L}(\theta^{s^t}, \theta^t) \]

MT Learning as Multi-Objective Optimization. Sener & Koltun. ‘19
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

T Yu, S Kumar, A Gupta, S Levine, K Hausman, C Finn. Gradient Surgery for Multi-Task Learning. '19
Idea: 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

T Yu, S Kumar, A Gupta, S Levine, K Hausman, C Finn. *Gradient Surgery for Multi-Task Learning.* '19
### Multi-Task CIFAR-100

<table>
<thead>
<tr>
<th>Task Description</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>task specific-1-fc (Rosenbaum et al., 2018)</td>
<td>42</td>
</tr>
<tr>
<td>task specific-all-fc (Rosenbaum et al., 2018)</td>
<td>49</td>
</tr>
<tr>
<td>cross stitch-all-fc (Misra et al., 2016b)</td>
<td>53</td>
</tr>
<tr>
<td>routing-all-fc + WPL (Rosenbaum et al., 2019) independent</td>
<td>74.7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>% Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCGrad (ours)</td>
<td>71</td>
</tr>
<tr>
<td>routing-all-fc + WPL + PCGrad (ours)</td>
<td>77.5</td>
</tr>
</tbody>
</table>

+ also helps multi-task **supervised** learning
+ complementary to multi-task **architectures**

### Multi-Task NYUv2

<table>
<thead>
<tr>
<th>#P</th>
<th>Architecture</th>
<th>Weighting</th>
<th>Segmentation</th>
<th>Depth</th>
<th>Surface Normal</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Higher Better) mIoU</td>
<td>(Lower Better) Abs Err Rel Err</td>
<td>Angle Distance (Lower Better) Mean</td>
</tr>
<tr>
<td>≈3</td>
<td>Cross-Stitch†</td>
<td>Equal Weights</td>
<td>14.71</td>
<td>50.23</td>
<td>0.6481</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uncert. Weights*</td>
<td>15.69</td>
<td>52.60</td>
<td>0.6277</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DWA†, $T = 2$</td>
<td>16.11</td>
<td>53.19</td>
<td><strong>0.5922</strong></td>
</tr>
<tr>
<td>1.77</td>
<td>MTAN†</td>
<td>Equal Weights</td>
<td>17.72</td>
<td>55.32</td>
<td><strong>0.5906</strong></td>
</tr>
<tr>
<td>1.77</td>
<td>MTAN† + PCGrad (ours)</td>
<td>Uncert. Weights*</td>
<td>17.67</td>
<td><strong>55.61</strong></td>
<td>0.5927</td>
</tr>
<tr>
<td>1.77</td>
<td></td>
<td>DWA†, $T = 2$</td>
<td>17.15</td>
<td>54.97</td>
<td>0.5956</td>
</tr>
</tbody>
</table>

T Yu, S Kumar, A Gupta, S Levine, K Hausman, C Finn. **Gradient Surgery for Multi-Task Learning.** ‘19
Why does it work?
(Part 1)

MT10

Success Rates vs. Number of thousand env steps

- SAC+PA+PCGrad
- SAC+PA+PCGrad dir
- SAC+PA+PCGrad mag

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?

T Yu, S Kumar, A Gupta, S Levine, K Hausman, C Finn. Gradient Surgery for Multi-Task Learning. ‘19
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

(for two tasks)

**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) $t \geq \frac{2}{\ell - \xi(g_1, g_2)L}$.

**Proof.** See Appendix B.

Are these three conditions actually why we see improvements on large-scale problems?
3. Can we scale meta-learning to broad task distributions?

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

Lack of good benchmarks —> **Meta-World** with broad, dense task distribution
scaling primarily hindered by *optimization* challenges in MTL

Optimization challenges —> three conditions seem to plague MTL, MTRL

**a solution**: project conflicting gradients (**PCGrad**)

**Remaining questions:**
Does this solution translate back to meta-learning?
Is this problem unique to multi-task learning?
2. A peculiar yet ubiquitous problem in meta-learning
   (and how we might regularize it away)

   **meta overfitting**
   memorize training functions $f_i$
corresponding to tasks in your meta-training dataset

   **meta regularization**
   controls information flow
regularizes description length
of meta-parameters

3. Can we scale meta-learning to broad task distributions?

   Lack of good benchmarks
   —> **Meta-World** with broad, dense task distribution
   scaling primarily hindered by *optimization* challenges in MTL

   Optimization challenges
   —> three conditions seem to plague MTL, MTRL
       **a solution**: project conflicting gradients (PCGrad)
Want to Learn More?

CS330: Deep Multi-Task & Meta-Learning
Lecture videos online!

Working on Meta-RL?

Try out the Meta-World benchmark

Collaborators

IRIS - RAIL retreat in Sonoma, CA

Yin, Tucker, Yuan, Levine, Finn. Meta-Learning without Memorization. ‘19
T Yu, D Quillen, Z He, R Julian, K Hausman, C Finn, S Levine. Meta-World. CoRL ‘19
T Yu, S Kumar, A Gupta, S Levine, K Hausman, C Finn. Gradient Surgery for Multi-Task Learning. ‘19