Non-Parametric Few-Shot Learning

CS 330
Course Reminders

Homework 1 due tonight.

Homework 2 released, due Weds 10/25.

Project mentors assigned: go to their office hours with any questions.

Project proposal due next Monday 10/23.
(graded lightly, for your benefit)

Following up on some feedback:
- Comparisons to simple baselines — included in today’s lecture
- Max Sobol Mark’s office hours (Weds 6-8 pm) moving to virtual
Plan for Today

Non-Parametric Few-Shot Learning
- Siamese networks, matching networks, prototypical networks

Properties of Meta-Learning Algorithms
- Comparison of approaches

Examples of Meta-Learning In Practice
- Imitation learning, drug discovery, motion prediction, language generation

Goals for by the end of lecture:
- Basics of non-parametric few-shot learning techniques (& how to implement)
- Trade-offs between black-box, optimization-based, and non-parametric meta-learning
- Familiarity with applied formulations of meta-learning

} Part of Homework 2!
Recap: **Black-Box Meta-Learning**

Key idea: parametrize learner as a neural network

+ **expressive**

- **challenging optimization** problem
Recap: Optimization-Based Meta-Learning

Key idea: embed optimization inside the inner learning process
Optimization-Based Adaptation

**Challenges.** Bi-level optimization can exhibit instabilities.

**Idea:** Automatically learn inner vector learning rate, tune outer learning rate
(Li et al. Meta-SGD, Behl et al. AlphaMAML)

**Idea:** Optimize only a subset of the parameters in the inner loop
(Zhou et al. DEML, Zintgraf et al. CAVIA)

**Idea:** Decouple inner learning rate, BN statistics per-step
(Antoniou et al. MAML++)

**Idea:** Introduce context variables for increased expressive power.
(Finn et al. bias transformation, Zintgraf et al. CAVIA)

**Takeaway:** a range of simple tricks that can help optimization significantly
Optimization-Based Adaptation

**Challenges.** Backpropagating through many inner gradient steps is compute- & memory-intensive.

**Idea:** [Crudely] approximate \( \frac{d\phi_i}{d\theta} \) as identity

(Finn et al. first-order MAML ‘17, Nichol et al. Reptile ‘18)

Surprisingly works for simple few-shot problems, but (anecdotally) not for more complex meta-learning problems.

**Idea:** Only optimize the *last layer* of weights.

- *Ridge regression, logistic regression* (Bertinetto et al. R2-D2 ’19)
- *Support vector machine* (Lee et al. MetaOptNet ’19)

\( \rightarrow \) leads to a *closed form* or *convex* optimization on top of meta-learned features

**Idea:** Derive meta-gradient using the implicit function theorem

(Rajeswaran, Finn, Kakade, Levine. Implicit MAML ’19)

\( \rightarrow \) compute full meta-gradient *without differentiating through optimization path*
Optimization-Based Adaptation

Key idea: Acquire $\phi_i$ through optimization.

**Takeaways:** Construct *bi-level optimization* problem.
+ positive inductive bias at the start of meta-learning
+ tends to extrapolate better via structure of optimization
+ maximally expressive with sufficiently deep network
+ model-agnostic (easy to combine with your favorite architecture)
- typically requires second-order optimization
- usually compute and/or memory intensive

-> Can be prohibitively expensive for large models
Recap: Optimization-Based Meta-Learning

Key idea: embed optimization inside the inner learning process

+ **structure** of optimization embedded into meta-learner
- memory-intensive, requires second-order optimization

**Today:** Can we embed a learning procedure *without* a second-order optimization?
So far: Learning parametric models.

In low data regimes, **non-parametric** methods are simple, work well.

During **meta-test time**: few-shot learning <-> low data regime
During **meta-training**: still want to be parametric

Can we use **parametric meta-learners** that produce effective **non-parametric learners**?

Note: some of these methods precede parametric approaches
Non-parametric methods

Key Idea: Use non-parametric learner.

Compare test image with training images

In what space do you compare? With what distance metric? $\ell_2$ distance in pixel space?
In what space do you compare? With what distance metric?

$\ell_2$ distance in pixel space?

Zhang et al. (arXiv 1801.03924)
Non-parametric methods

**Key Idea:** Use non-parametric learner.

Compare test image with training images

**In what space do you compare? With what distance metric?**

$\ell_2$ distance in pixel space?

**Question:** What distance metric would you use instead?

**Idea:** Learn to compare using meta-training data
Non-parametric methods

**Key Idea**: Use non-parametric learner.

train Siamese network to predict whether or not two images are the same class
Non-parametric methods

**Key Idea:** Use non-parametric learner.

train Siamese network to predict whether or not two images are the same class

Koch et al., ICML ‘15
Non-parametric methods

**Key Idea:** Use non-parametric learner.

train Siamese network to predict whether or not two images are the same class

Koch et al., ICML ‘15
Non-parametric methods

**Key Idea:** Use non-parametric learner.

train Siamese network to predict whether or not two images are the same class

Meta-test time: compare image $\mathbf{X}_{\text{test}}$ to each image in $\mathcal{D}_{\text{tr}}$

**Meta-training:** Binary classification

**Meta-test:** N-way classification

Can we **match** meta-train & meta-test?
Non-parametric methods

**Key Idea:** Use non-parametric learner.

Can we **match** meta-train & meta-test?

Nearest neighbors in learned embedding space

\[ y_{ts} = \sum_{x_k, y_k \in D_{tr}} f_\theta(x_{ts}, x_k) y_k \]

Trained end-to-end.

Meta-train & meta-test time match.
Non-parametric methods

**Key Idea:** Use non-parametric learner.

**General Algorithm:**

1. Sample task $T_i$ (or mini batch of tasks)
2. Sample disjoint datasets $D_{i}^{tr}, D_{i}^{test}$ from $D_i$
3. Compute $\phi_i \leftarrow f_\theta(D_{i}^{tr})$ Compute $\hat{y}^{ts} = \sum_{x_k, y_k \in D_{i}^{tr}} f_\theta(x^{ts}, x_k) y_k$
4. Update $\theta$ using $\nabla_\theta \mathcal{L}(\phi_i, D_{i}^{test})$ Update $\theta$ using $\nabla_\theta \mathcal{L}(\hat{y}^{ts}, y^{ts})$

**Black-box approach** — Non-parametric approach (matching networks)

What if >1 shot?

Matching networks will perform comparisons independently

Can we aggregate class information to create a prototypical embedding?
Non-parametric methods

**Key Idea:** Use non-parametric learner.

\[ c_n = \frac{1}{K} \sum_{(x,y) \in D_{i}^\text{tr}} \mathbb{1}(y = n) f_\theta(x) \]

\[ p_\theta(y = n|x) = \frac{\exp(-d(f_\theta(x), c_n))}{\sum_{n'} \exp(-d(f_\theta(x), c_{n'}))} \]

\( d: \) Euclidean, or cosine distance
Non-parametric methods

So far: Siamese networks, matching networks, prototypical networks
Embed, then nearest neighbors.

Challenge
What if you need to reason about more complex relationships between datapoints?

Idea: Learn non-linear relation module on embeddings
Sung et al. Relation Net ‘17

Idea: Learn infinite mixture of prototypes.
Allen et al. IMP, ICML ‘19

Idea: Perform message passing on embeddings
Garcia & Bruna, GNN ‘17
Plan for Today

Non-Parametric Few-Shot Learning
- Siamese networks, matching networks, prototypical networks

Properties of Meta-Learning Algorithms
- Comparison of approaches

Examples of Meta-Learning In Practice
- Imitation learning, drug discovery, motion prediction, language generation

How can we think about how these methods compare?
Black-box vs. Optimization vs. Non-Parametric

**Computation graph perspective**

**Black-box**

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

**Optimization-based**

\[ y^{ts} = f_{\text{MAML}}(D^{tr}_i, x^{ts}) = f_{\phi_i}(x^{ts}) \]

where \( \phi_i = \theta - \alpha \nabla_\theta \mathcal{L}(\theta, D^{tr}_i) \)

**Non-parametric**

\[ y^{ts} = f_{\text{PN}}(D^{tr}_i, x^{ts}) = \text{softmax}(-d(f_\theta(x^{ts}), c_n)) \]

where \( c_n = \frac{1}{K} \sum_{(x,y) \in D^{tr}_i} \mathbb{1}(y = n)f_\theta(x) \)

Note: (again) Can mix & match components of computation graph

Gradient descent on relation net embedding.

Both condition on data & run gradient descent.

Jiang et al. CAML ‘19

MAML, but initialize last layer as ProtoNet during meta-training

Triantafillou et al. Proto-MAML ‘19

Rusu et al. LEO ‘19
Black-box vs. Optimization vs. Non-Parametric

**Algorithmic properties** perspective

Expressive power

- the ability for $f$ to represent a range of learning procedures
  
  *Why?* scalability, applicability to a range of domains

Consistency

- learned learning procedure will monotonically improve with more data
  
  *Why?* reduce reliance on meta-training tasks, good OOD task performance

Recall:

These properties are important for most applications!
Black-box vs. Optimization vs. Non-Parametric

**Black-box**
- + complete expressive power
- - not consistent
- + easy to combine with variety of learning problems (e.g. SL, RL)
- - challenging optimization (no inductive bias at the initialization)
- - often data-inefficient

**Optimization-based**
- + consistent, reduces to GD
- ~ expressive for very deep models*
- + positive inductive bias at the start of meta-learning
- + handles varying & large K well
- - second-order optimization
- - compute and memory intensive

**Non-parametric**
- + expressive for most architectures
- ~ consistent under certain conditions
- + entirely feedforward
- + computationally fast & easy to optimize
- - harder to generalize to varying K
- - hard to scale to very large K
- - so far, limited to classification

Generally, well-tuned versions of each perform **comparably** on many few-shot benchmarks! (likely says more about the benchmarks than the methods)
Which method to use depends on your **use-case**.

*for supervised learning settings
Black-box vs. Optimization vs. Non-Parametric

**Algorithmic properties perspective**

Expressive power

- the ability for $f$ to represent a range of learning procedures
  - **Why?** scalability, applicability to a range of domains

Consistency

- learned learning procedure will monotonically improve with more data
  - **Why?** reduce reliance on meta-training tasks, good OOD task performance

Uncertainty awareness

- ability to reason about ambiguity during learning
  - **Why?** active learning, calibrated uncertainty, RL principled Bayesian approaches

**We’ll discuss this in 2 weeks!**
Plan for Today

Non-Parametric Few-Shot Learning
- Siamese networks, matching networks, prototypical networks

Properties of Meta-Learning Algorithms
- Comparison of approaches

Examples of Meta-Learning In Practice
- Imitation learning, drug discovery, motion prediction, language generation
Application: Land-Cover Classification

(Rubwurm*, Wang* et al. Meta-Learning for Few-Shot Land-Cover Classification. CVPR EarthVision 2020)

**Tasks:**
Classification or segmentation of image
Different regions of the world
\( \mathcal{D}_i^{tr}, \mathcal{D}_i^{ts} \): images from a particular region

**Model:** optimization-based (MAML)

---

**SEN12MS dataset**
(Schmitt et al. 2019)
Application: Student Feedback Generation

(Wu et al. Prototransformer: A meta-learning approach to providing student feedback. 2021)

Tasks:
Different rubric items from different exams
\( D_i^{tr}, D_i^{ts} \): student solutions (python programs)

Model: non-parametric
Protonets with pre-trained transformer, task augmentation, side information
### Main Offline Results

<table>
<thead>
<tr>
<th>Model</th>
<th>AP</th>
<th>Held-out rubric</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P@50</td>
<td>P@75</td>
<td>ROC-AUC</td>
</tr>
<tr>
<td>ProtoTransformer</td>
<td>84.2</td>
<td>85.2</td>
<td>74.2</td>
<td>82.9</td>
</tr>
<tr>
<td></td>
<td>(±1.7)</td>
<td>(±3.8)</td>
<td>(±1.4)</td>
<td>(±1.3)</td>
</tr>
<tr>
<td>Supervised</td>
<td>66.9</td>
<td>59.1</td>
<td>53.9</td>
<td>61.0</td>
</tr>
<tr>
<td></td>
<td>(±2.2)</td>
<td>(±1.7)</td>
<td>(±1.5)</td>
<td>(±2.1)</td>
</tr>
<tr>
<td>Human TA</td>
<td>82.5</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>

- **Supervised baseline**: train classifier per task, using same pre-trained CodeBERT

- Outperforms supervised learning by **8-17%**

- More accurate than human TA on held-out rubric

- **Room to grow** on held-out exam

<table>
<thead>
<tr>
<th>Model</th>
<th>AP</th>
<th>Held-out exam</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>P@50</td>
<td>P@75</td>
<td>ROC-AUC</td>
</tr>
<tr>
<td>ProtoTransformer</td>
<td>74.2</td>
<td>77.3</td>
<td>67.3</td>
<td>77.0</td>
</tr>
<tr>
<td></td>
<td>(±1.6)</td>
<td>(±2.7)</td>
<td>(±2.0)</td>
<td>(±1.4)</td>
</tr>
<tr>
<td>Supervised</td>
<td>65.8</td>
<td>60.1</td>
<td>54.3</td>
<td>60.7</td>
</tr>
<tr>
<td></td>
<td>(± 2.1)</td>
<td>(±3.0)</td>
<td>(±1.8)</td>
<td>(±1.6)</td>
</tr>
<tr>
<td>Human TA</td>
<td>82.5</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
</tbody>
</table>
Application: Low-Resource Molecular Property Prediction

(Nguyen et al. Meta-Learning GNN Initializations for Low-Resource Molecular Property Prediction. 2020)

[potentially useful for low-resource drug discovery problems]

**Tasks:**

Predicting properties & activities of different molecules

\( \mathcal{D}_i^{tr}, \mathcal{D}_i^{ts} \): different instances

**Model:** optimization-based
MAML, first-order MAML, ANIL
Gated graph neural net base model

<table>
<thead>
<tr>
<th>CHEMBL ID</th>
<th>k-NN</th>
<th>FINETUNE-ALL</th>
<th>FINETUNE-Top</th>
<th>FO-MAML</th>
<th>ANIL</th>
<th>MAML</th>
</tr>
</thead>
<tbody>
<tr>
<td>2363236</td>
<td>0.316 ± 0.007</td>
<td>0.328 ± 0.028</td>
<td>0.329 ± 0.023</td>
<td>0.337 ± 0.019</td>
<td>0.325 ± 0.008</td>
<td>0.332 ± 0.013</td>
</tr>
<tr>
<td>1614469</td>
<td>0.438 ± 0.023</td>
<td>0.470 ± 0.034</td>
<td>0.480 ± 0.033</td>
<td>0.489 ± 0.019</td>
<td>0.446 ± 0.044</td>
<td>0.507 ± 0.030</td>
</tr>
<tr>
<td>2363146</td>
<td>0.559 ± 0.026</td>
<td>0.626 ± 0.037</td>
<td>0.653 ± 0.029</td>
<td>0.555 ± 0.017</td>
<td>0.506 ± 0.034</td>
<td>0.595 ± 0.051</td>
</tr>
<tr>
<td>2363366</td>
<td>0.511 ± 0.059</td>
<td>0.557 ± 0.039</td>
<td>0.561 ± 0.048</td>
<td>0.546 ± 0.037</td>
<td>0.570 ± 0.051</td>
<td>0.598 ± 0.041</td>
</tr>
<tr>
<td>2363553</td>
<td>0.739 ± 0.007</td>
<td>0.724 ± 0.015</td>
<td>0.737 ± 0.023</td>
<td>0.694 ± 0.011</td>
<td>0.686 ± 0.020</td>
<td>0.691 ± 0.013</td>
</tr>
<tr>
<td>1963818</td>
<td>0.607 ± 0.041</td>
<td>0.708 ± 0.036</td>
<td>0.595 ± 0.142</td>
<td>0.677 ± 0.026</td>
<td>0.692 ± 0.061</td>
<td>0.745 ± 0.048</td>
</tr>
<tr>
<td>1963945</td>
<td>0.805 ± 0.031</td>
<td>0.848 ± 0.034</td>
<td>0.835 ± 0.036</td>
<td>0.779 ± 0.039</td>
<td>0.753 ± 0.033</td>
<td>0.836 ± 0.023</td>
</tr>
<tr>
<td>1614423</td>
<td>0.503 ± 0.044</td>
<td>0.628 ± 0.058</td>
<td>0.642 ± 0.063</td>
<td>0.760 ± 0.024</td>
<td>0.730 ± 0.077</td>
<td>0.837 ± 0.036</td>
</tr>
<tr>
<td>2114825</td>
<td>0.679 ± 0.027</td>
<td>0.739 ± 0.050</td>
<td>0.732 ± 0.051</td>
<td>0.837 ± 0.042</td>
<td>0.759 ± 0.078</td>
<td>0.885 ± 0.014</td>
</tr>
<tr>
<td>1964116</td>
<td>0.709 ± 0.042</td>
<td>0.758 ± 0.044</td>
<td>0.769 ± 0.048</td>
<td>0.895 ± 0.023</td>
<td>0.903 ± 0.012</td>
<td>0.912 ± 0.013</td>
</tr>
<tr>
<td>2155446</td>
<td>0.471 ± 0.008</td>
<td>0.473 ± 0.017</td>
<td>0.476 ± 0.013</td>
<td>0.497 ± 0.024</td>
<td>0.478 ± 0.020</td>
<td>0.500 ± 0.017</td>
</tr>
<tr>
<td>1909204</td>
<td>0.539 ± 0.023</td>
<td>0.559 ± 0.031</td>
<td>0.577 ± 0.039</td>
<td>0.592 ± 0.043</td>
<td>0.547 ± 0.029</td>
<td>0.601 ± 0.027</td>
</tr>
<tr>
<td>1909213</td>
<td>0.694 ± 0.009</td>
<td>0.742 ± 0.015</td>
<td>0.759 ± 0.012</td>
<td>0.698 ± 0.024</td>
<td>0.694 ± 0.025</td>
<td>0.729 ± 0.013</td>
</tr>
<tr>
<td>3111197</td>
<td>0.617 ± 0.028</td>
<td>0.663 ± 0.066</td>
<td>0.673 ± 0.071</td>
<td>0.636 ± 0.036</td>
<td>0.737 ± 0.035</td>
<td>0.746 ± 0.045</td>
</tr>
<tr>
<td>3215171</td>
<td>0.480 ± 0.042</td>
<td>0.552 ± 0.043</td>
<td>0.561 ± 0.045</td>
<td>0.729 ± 0.031</td>
<td>0.700 ± 0.060</td>
<td>0.764 ± 0.019</td>
</tr>
<tr>
<td>3215034</td>
<td>0.474 ± 0.072</td>
<td>0.540 ± 0.156</td>
<td>0.455 ± 0.189</td>
<td>0.819 ± 0.048</td>
<td>0.681 ± 0.042</td>
<td>0.805 ± 0.046</td>
</tr>
<tr>
<td>1909103</td>
<td>0.881 ± 0.026</td>
<td>0.936 ± 0.013</td>
<td>0.921 ± 0.020</td>
<td>0.877 ± 0.046</td>
<td>0.736 ± 0.055</td>
<td>0.900 ± 0.032</td>
</tr>
<tr>
<td>3215092</td>
<td>0.696 ± 0.038</td>
<td>0.771 ± 0.039</td>
<td>0.791 ± 0.044</td>
<td>0.877 ± 0.028</td>
<td>0.801 ± 0.026</td>
<td>0.907 ± 0.017</td>
</tr>
<tr>
<td>1753253</td>
<td>0.710 ± 0.084</td>
<td>0.850 ± 0.029</td>
<td>0.861 ± 0.025</td>
<td>0.885 ± 0.033</td>
<td>0.758 ± 0.114</td>
<td>0.908 ± 0.011</td>
</tr>
<tr>
<td>1614549</td>
<td>0.710 ± 0.035</td>
<td>0.850 ± 0.041</td>
<td>0.860 ± 0.051</td>
<td>0.930 ± 0.022</td>
<td>0.906 ± 0.050</td>
<td>0.947 ± 0.014</td>
</tr>
</tbody>
</table>

**AVG. RANK**: 5.4, 3.5, 3.5, 3.1, 4.0, 1.7
Side note

$\mathcal{D}_{i}^{tr}$ and $\mathcal{D}_{i}^{ts}$ do not need to be sampled independently from $\mathcal{D}_{i}$.

$\mathcal{D}_{i}^{tr}$ could have:
- noisy labels
- weakly supervised
- domain shift
- etc.
Application: One-Shot Imitation Learning

(Yu*, Finn* et al. One-Shot Imitation from Observing Humans. RSS 2018)

Tasks:
- manipulating different objects
- $\mathcal{D}_{tr}^i$: video of a human
- $\mathcal{D}_{ts}^i$: teleoperated demonstration

Model: optimization-based
- MAML with learned inner loss
Application: Dermatological Image Classification
(Prabhu et al. Prototypical Clustering Networks for Dermatological Image Classification. ML4HC 2019)

**Tasks:**
Different skin conditions
\( \mathcal{D}_i^{tr}, \mathcal{D}_i^{ts} \): images from different people

**Goal:** good classifier on all classes.

**Model:** non-parametric
Protonets, multiple prototypes per class using clustering objective
**Evaluation**

**Compare:**

- **PN** - standard ProtoNets, trained on 150 base classes, pre-trained on ImageNet
- **FT\textsubscript{N}-1NN** - ImageNet pre-training, fine-tuned ResNet on \(N\) classes, 1-nearest neighbors in resulting embedding space
- **FT\textsubscript{200}-CE** - ImageNet pre-trained, fine-tuned on all 200 classes with balancing (very strong baseline, accesses more info during training, requires re-training for new classes)

**Evaluation Metric:** mean class accuracy (mca), i.e. average of per-class accuracies across 200 classes.

<table>
<thead>
<tr>
<th>Approach</th>
<th>(mca\textsubscript{base} + \text{novel})</th>
<th>(k = 5)</th>
<th>(k = 10)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(FT\textsubscript{150-1NN})</td>
<td>46.18 +/- 0.81</td>
<td>55.32 +/- 0.30</td>
<td>18.76 +/- 3.30</td>
</tr>
<tr>
<td>(FT\textsubscript{150-3NN})</td>
<td>44.28 +/- 0.32</td>
<td>54.77 +/- 0.47</td>
<td>12.80 +/- 1.50</td>
</tr>
<tr>
<td>(FT\textsubscript{200-1NN})</td>
<td>46.52 +/- 0.39</td>
<td>54.17 +/- 0.30</td>
<td>22.50 +/- 0.75</td>
</tr>
<tr>
<td>(FT\textsubscript{200-3NN})</td>
<td>44.69 +/- 0.39</td>
<td>52.61 +/- 0.21</td>
<td>20.93 +/- 2.00</td>
</tr>
<tr>
<td>(FT\textsubscript{200-CE})</td>
<td>47.82 +/- 0.46</td>
<td>55.75 +/- 0.71</td>
<td>24.00 +/- 3.22</td>
</tr>
<tr>
<td><strong>PN</strong></td>
<td>43.92 +/- 0.40</td>
<td>48.71 +/- 0.37</td>
<td>29.56 +/- 2.35</td>
</tr>
<tr>
<td><strong>PCN (ours)</strong></td>
<td><strong>47.79 +/- 0.71</strong></td>
<td>53.70 +/- 0.18</td>
<td><strong>30.04 +/- 2.77</strong></td>
</tr>
</tbody>
</table>

**PCN > PN**

**PCN > FT\textsubscript{N}-*NN**

**PCN \approx FT\textsubscript{200-CE}** without requiring re-training

More visualizations and analysis in the paper!
Application: Few-Shot Human Motion Prediction

(Gui et al. Few-Shot Human Motion Prediction via Meta-Learning. ECCV 2018)
[potentially useful for human-robot interaction, autonomous driving]

**Tasks:**
Different human users & motions

$\mathcal{D}^\text{tr}_i$: past K time steps of motion

$\mathcal{D}^\text{ts}_i$: future second(s) of motion

**Model:**
optimization-based/black-box hybrid

MAML with additional learned update rule

Recurrent neural net base model

Mean angle error w.r.t. prediction horizon

<table>
<thead>
<tr>
<th>GT</th>
<th>...</th>
<th>PAML</th>
<th>...</th>
</tr>
</thead>
</table>

### Table 1: Performance Comparison

<table>
<thead>
<tr>
<th></th>
<th>Walking</th>
<th>Eating</th>
<th>Smoking</th>
<th>Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>milliseconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>160</td>
<td>320</td>
<td>400</td>
</tr>
<tr>
<td>residual sup. [32] w/ (Baselines)</td>
<td>Scratch$_{pec}$</td>
<td>Scratch$_{an}$</td>
<td>Transfer$_{ets}$</td>
<td>Multi-task</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Walking</th>
<th>Eating</th>
<th>Smoking</th>
<th>Discussion</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>milliseconds</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>80</td>
<td>160</td>
<td>320</td>
<td>400</td>
</tr>
<tr>
<td>residual sup. [32] w/ (Baselines)</td>
<td>Scratch$_{pec}$</td>
<td>Scratch$_{an}$</td>
<td>Transfer$_{ets}$</td>
<td>Multi-task</td>
</tr>
</tbody>
</table>
Plan for Today

Non-Parametric Few-Shot Learning
- Siamese networks, matching networks, prototypical networks

Properties of Meta-Learning Algorithms
- Comparison of approaches

Examples of Meta-Learning in Practice
- Imitation learning, drug discovery, motion prediction, language generation

Goals for by the end of lecture:
- Basics of non-parametric few-shot learning techniques (& how to implement)
- Trade-offs between black-box, optimization-based, and non-parametric meta-learning
- Familiarity with applied formulations of meta-learning

} Part of Homework 2!
Course Reminders

Done with meta-learning algorithms!

**Lectures**

Next: unsupervised pre-training

**Homeworks**

Homework 1 due **tonight**.

Homework 2 released, due Weds 10/25.

**Project**

Project mentors assigned: go to their OH with questions.

Project proposal due next Monday 10/23.

(graded lightly, for your benefit)