Soft Weight Networks for Few Shot Learning

Presenter:
Gregory Ditzler, The University of Arizona

Project Participants:
Samuel Hess, The University of Arizona
Arianna Pryor, The University of Arizona
Carmelo Moraila, The University of Arizona
Outline

- Project Overview
  - Goals, Motivation, Challenges
  - Team Members
  - Background and Related Tasks

- Project Status
  - Tasks
  - Deliverables and Benefits
  - Prior Work and Results
  - Activities and Outcome

- Project Resources and Finances
  - Current status
  - Plans from now to completion

- Conclusions
Few shot learning is a machine learning task that is very easy for humans, but difficult to achieve with a neural network.

Classical Machine Learning (ML) paradigm

- Given a dataset, run a ML algorithm to build a model and make predictions on unseen data.

In many real-world application data are provided over time and old data are either not available or easy to access.

In this project, we propose to develop deep neural networks and sophisticated learning (not traditional) to achieve lifelong machine learning and integrate few-shot into the architecture.
This is a serious problem in machine learning and this project seeks to address new classes learning over time.
Project Goals, Motivation, Challenges

**Project Goals**

- Develop deep neural network learning strategies that can learn in setting where new classes are presented then automatically identify the same class of samples
- Leverage multiple GPUs to speed up the training process of our algorithms and the algorithms we are going to benchmark against

**Motivations**

- Learning new classes with too little data is a highly nontrivial task for a neural network to perform without overtraining
- Lifelong learning is difficult to achieve with a neural network without catastrophic forgetting
- The settings we are proposing to address are what humans encounter

**Challenges**

- Training our deep neural networks take hours to 8+ days to train
- Moving to the online model’s training without catastrophic forgetting is still an open problem
Project Team Members

- **Faculty**
  - Gregory Ditzler, Ph.D.
    Assistant Professor, University of Arizona

- **Students**
  - Samuel Hess
    Ph.D. Candidate, University of Arizona
  - Arianna Pryor
    Ph.D. Candidate, University of Arizona
  - Carmelo Moraila
    Masters Student, University of Arizona

*This project leverages collaborative work done with students at all levels, and is at the intersection of machine learning and computing*
Overview

- Goal: Given only one (or “few”) labeled examples, correctly identify other unlabeled examples with arbitrarily high accuracy.

### 20-Way 1-Shot Example

**Given**
20 one-shot labeled sample

**Problem**
label this sample

**Answer**
example belongs to label 6

Prior work: Train with episodes of randomly selected few-shot batches to establish feature embedded “prototypes”

**Soft Weight Network Approach:** Use randomly selected batches to learn feature embedding AND weightings for cross comparison of labeled samples and test query during runtime
### Background and Related Research

#### Few-Shot Learning
Given only one (or “few”) labeled examples, correctly identify other unlabeled examples with arbitrarily high accuracy.

#### Online Learning
As new classes are appear in the testing data be able to identify and classify new and old classes without retraining on the full dataset.

#### Incremental Learning
As new classes are incrementally added to training data be able to classify new and old classes without retraining on the full dataset.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Few-shot</th>
<th>Incremental</th>
<th>Online</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low # of class samples</td>
<td>YES</td>
<td>NO</td>
<td>NO</td>
</tr>
<tr>
<td>Low # of classes</td>
<td>NO</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Bounded memory</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Data stream</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
<tr>
<td>Identify new class</td>
<td>NO</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Forgetting</td>
<td>YES</td>
<td>NO</td>
<td>YES</td>
</tr>
<tr>
<td>Competitive multi-class performance</td>
<td>YES</td>
<td>YES</td>
<td>YES</td>
</tr>
</tbody>
</table>
Overview of Project Tasks

- **(1) SWNs for large-scale data sets**
  - Finalize benchmarks on the full ImageNet data set
  - Submit a journal

- **(2) SWNs+Lifelong Learning**
  - Develop a novel learning strategy that can achieve lifelong learning with few-shot, while minimizing catastrophic forgetting
  - Prepare a conference paper and journal manuscript
  - Implement novelty detection for new classes

- **(3) Large-scale implementation of (3) and Benchmarks**
  - Identify data sets and algorithms to benchmark
Research Questions

- How do we effectively and efficiently train an SWN with a low # of classes but high # of samples/class?
  - Will the large number of samples/class account for the lack of classes?
- How can we identify novel classes using few-shot?
- Lifelong learning requires that we dynamically adjust network weights over time to compensate for distribution drift or impurities in the data. How can this task be achieved with lifelong learning without catastrophic forgetting?
- How might a consolidation or “sleep” phase work with a few-shot training algorithm?
  - FearNet evaluates a few-shot network but does not train the few-shot network in the way that worked well for matching nets.
Deliverables and benefits

- **Deliverables**
  - Submit a manuscript to the IEEE Transactions on Neural Networks and Learning Systems (Impact Factor: 7.982)
  - Planned conference papers
    - IEEE/INNS International Joint Conference on Neural Networks
    - IEEE International Conference Acoustics, Speech and Signal Processing
  - Open source software for all implementations (Github)

- **Benefits**
  - Advance the science of few-shot learning and lifelong learning
  - Develop a better understanding on how to achieve more intelligent and effective forms of learning deep networks
Prior Work and Results
Prior Work and Results

**Algorithm 1 Soft Weight Network pseudo-code**

**Input:** Entire Training Set $Tr = \{(x_1, y_1), \ldots, (x_{N_{tr}}, y_{N_{tr}})\}$, number of support samples per episode $N$, number of query samples per episode $M$, number of classes per episode $C$, initial model $\phi$, and model learning rate $\eta$

**Output:** updated model $\phi$, network loss $J(\phi)$, and class estimates $\tilde{y}_{qm}$

1. for each episode in training data do
2. Create a set, $E = \{(x_1, y_1), \ldots, (x_{N_{TE}}, y_{N_{TE}})\}$, of $C$ different classes randomly chosen from $Tr$ without replacement
3. Create disjoint sets, $S = \{(x_1, y_1), \ldots, (x_N, y_N)\}$ and $Q = \{(x_1, y_1), \ldots, (x_M, y_M)\}$, of $N$ support examples and $M$ query samples from $E$
4. for $q = 1, \ldots, M$ do // Loop through query samples
5. for $c = 1, \ldots, C$ do // Loop through class samples
6. // Compute the classification scores for every query sample
7. $\alpha_{qm}(c) = \sum_{(x_{s}, y_{s}) \in S_c} \left[ h_\phi(f_\phi(x_{s})) + f_\phi^T(x_{qm}) w_\phi(f_\phi(x_{s})) \right]$
8. $\beta_{qm}(c) = \sum_{(x_{s}, y_{s}) \in S_c} \left[ h_\phi(f_\phi(x_{qm})) + f_\phi^T(x_{s}) w_\phi(f_\phi(x_{qm})) \right]$
9. end for
10. end for
11. for $q = 1, \ldots, M$ do // Estimate posterior via softmax
12. for $c = 1, \ldots, C$ do // Estimate class of query sample
13. $p_\phi(y = c|x_{qm}) = \frac{e^{(\alpha_{qm}(c) + \beta_{qm}(c))}}{\sum_{c' \in C} e^{(\alpha_{qm}(c') + \beta_{qm}(c'))}}$
14. end for
15. $\tilde{y}_{qm} = \arg \max_{c \in C} [p_\phi(y = c|x_{qm})]$
16. end for
17. $J(\phi) = -\frac{1}{CM} \sum_{c'=1}^{C} \sum_{q=1}^{M} \log(p_\phi(y = c'|x_{qm}))$
18. $\phi \leftarrow \phi - \eta \nabla_\phi J(\phi)$ // Update model
19. end for

---

### Comparison of SWN and Protonet Performance on MiniImageNet

<table>
<thead>
<tr>
<th>Model</th>
<th>Classification Performance 1-Shot</th>
</tr>
</thead>
<tbody>
<tr>
<td>Soft Weight Networks</td>
<td>99.7%</td>
</tr>
<tr>
<td>Prototypical Networks* [12]</td>
<td>98.8%</td>
</tr>
<tr>
<td>Prototypical Networks with expansion</td>
<td>98.2%</td>
</tr>
<tr>
<td>Soft Weight Networks with reduction</td>
<td>99.7%</td>
</tr>
<tr>
<td>Soft Weight Networks Euclidean distance</td>
<td>99.4%</td>
</tr>
</tbody>
</table>

* Results reported by the authors.
## Prior Work and Results

### TABLE I

**OMNIGLOT PERFORMANCE**

<table>
<thead>
<tr>
<th>Model</th>
<th>Classification Performance</th>
<th>5-Way</th>
<th>20-Way</th>
<th>5-Way</th>
<th>20-Way</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siamese* [9]</td>
<td></td>
<td>98.8%</td>
<td>95.5%</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Matching Networks*  [11]</td>
<td></td>
<td>98.1%</td>
<td>93.8%</td>
<td>98.9%</td>
<td>98.5%</td>
</tr>
<tr>
<td>Prototypical Networks* [12]</td>
<td></td>
<td>98.8%</td>
<td>96.0%</td>
<td>99.7%</td>
<td>98.9%</td>
</tr>
<tr>
<td>MAML* [7]</td>
<td></td>
<td>98.7%</td>
<td>95.8%</td>
<td>99.9%</td>
<td>98.9%</td>
</tr>
<tr>
<td>ConvNet w/Memory* [17]</td>
<td></td>
<td>98.4%</td>
<td>95.0%</td>
<td>99.6%</td>
<td>98.6%</td>
</tr>
<tr>
<td>mAP-SSVM* [18]</td>
<td></td>
<td>98.6%</td>
<td>95.2%</td>
<td>99.6%</td>
<td>98.6%</td>
</tr>
<tr>
<td>mAP-DLM* [18]</td>
<td></td>
<td>98.8%</td>
<td>95.4%</td>
<td>99.6%</td>
<td>98.6%</td>
</tr>
<tr>
<td>Soft Weight Networks</td>
<td></td>
<td><strong>99.7%</strong></td>
<td><strong>98.3%</strong></td>
<td><strong>99.9%</strong></td>
<td><strong>99.6%</strong></td>
</tr>
</tbody>
</table>

* Results reported by the authors.

### TABLE II

**MINIIMAGE-NET PERFORMANCE**

<table>
<thead>
<tr>
<th>Model</th>
<th>Classification Performance</th>
<th>1-Shot; 5-Way</th>
<th>5-Shot; 5-Way</th>
</tr>
</thead>
<tbody>
<tr>
<td>Siamese* [9]</td>
<td></td>
<td>48.42 ± 0.79%</td>
<td>–</td>
</tr>
<tr>
<td>Matching Nets * [11]</td>
<td></td>
<td>43.40 ± 0.78%</td>
<td>51.09 ± 0.71%</td>
</tr>
<tr>
<td>Matching Nets FCE* [11]</td>
<td></td>
<td>43.56 ± 0.84%</td>
<td>55.31 ± 0.73%</td>
</tr>
<tr>
<td>Prototypical Nets* [12]</td>
<td></td>
<td>49.42 ± 0.78%</td>
<td><strong>68.20 ± 0.66%</strong></td>
</tr>
<tr>
<td>MAML* [7]</td>
<td></td>
<td>48.70 ± 1.84%</td>
<td>63.11 ± 0.92%</td>
</tr>
<tr>
<td>Meta-learner LSTM* [5]</td>
<td></td>
<td>43.44 ± 0.77%</td>
<td>60.60 ± 0.71%</td>
</tr>
<tr>
<td>mAP-SSVM* [18]</td>
<td></td>
<td><strong>50.32 ± 0.80%</strong></td>
<td>63.94 ± 0.72%</td>
</tr>
<tr>
<td>mAP-DLM* [18]</td>
<td></td>
<td><strong>50.28 ± 0.80%</strong></td>
<td>63.70 ± 0.70%</td>
</tr>
<tr>
<td>Soft Weight Nets</td>
<td></td>
<td><strong>51.59 ± 0.73%</strong></td>
<td>66.87 ± 0.61%</td>
</tr>
<tr>
<td>Soft Weight Nets</td>
<td></td>
<td><strong>53.67 ± 0.73%</strong></td>
<td><strong>69.83 ± 0.63%</strong></td>
</tr>
</tbody>
</table>

### TABLE III

**COMPARISON OF PROTOTYPICAL AND SOFT WEIGHT NETWORK CONFIGURATIONS ON OMNIGLOT DATA SET**

<table>
<thead>
<tr>
<th>Model</th>
<th>Classification Performance</th>
<th>1-Shot</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>5-Way</td>
</tr>
<tr>
<td>Soft Weight Networks</td>
<td></td>
<td>99.7%</td>
</tr>
<tr>
<td>Prototypical Networks* [12]</td>
<td></td>
<td>98.8%</td>
</tr>
<tr>
<td>Prototypical Networks with expansion</td>
<td></td>
<td>98.2%</td>
</tr>
<tr>
<td>Soft Weight Networks with reduction</td>
<td></td>
<td>99.7%</td>
</tr>
<tr>
<td>Soft Weight Networks Euclidean distance</td>
<td></td>
<td>99.4%</td>
</tr>
</tbody>
</table>

* Results reported by the authors.
Prior Work and Results
Activities and outcomes

- **Activities**
  - Q1: Demonstrate the feasibility of SWNs on standard benchmark data sets against the state-of-the-art (in progress)
  - Q2: Develop a multi-GPU implementation for SWN to scale to the full ImageNet data set
  - Q3: Research and design a lifelong learning framework that incorporates few-shot learning
  - Q4: Implement a multi-GPU implementation from the network in Q3 on ImageNet and benchmark against the state-of-the-art

- **Outcomes**
  - Advance the science of machine learning to achieve a more natural approach to learning
  - Open source code will be made publicly available
Conclusions

Few-shot learning is a very challenging task with deep neural networks and training these networks is computationally very difficult.

This project seeks to extend our soft weight neural network to lifelong learning scenarios

The PI’s group is supported by an Nvidia Hardware Grant
Please take a moment to fill out your L.I.F.E. forms.

http://www.iucrc.com
Select “Cloud and Autonomic Computing Center” then select “IAB” role.

What do you like about this project?
What would you change?
(Please include all relevant feedback.)