Lifelong Learning

CS 330
Reminders

This Wednesday:

Lecture and instructor office hours over zoom

Tuesday (Nov 30th):

Project poster session

Wednesday (Dec 8th):

Project due
Plan for Today

The lifelong learning problem statement

Basic approaches to lifelong learning

Can we do better than the basics?

Revisiting the problem statement from the meta-learning perspective
A brief review of problem statements.

**Multi-Task Learning**
Learn to solve a set of tasks.

**Meta-Learning**
Given i.i.d. task distribution, learn a new task efficiently.
In contrast, many real world settings look like:

Our agents may not be given a large batch of data/tasks right off the bat!

Some examples:
- a student learning concepts in school
- a deployed image classification system learning from a stream of images from users
- a robot acquiring an increasingly large set of skills in different environments
- a virtual assistant learning to help different users with different tasks at different points in time
- a doctor’s assistant aiding in medical decision-making
Some Terminology

Sequential learning settings

- online learning
- lifelong learning
- continual learning
- incremental learning
- streaming data

distinct from sequence data and sequential decision-making
What is the lifelong learning *problem statement*?

**Exercise:**

1. Pick an example setting.

2. Discuss problem statement in your break-out room:
   
   (a) how would you set-up an experiment to develop & test your algorithm?
   
   (b) what are desirable/required properties of the algorithm?
   
   (c) how do you evaluate such a system?

---

**Example settings:**

A. a **student** learning concepts in school

B. a deployed **image classification system** learning from a stream of images from users

C. a **robot** acquiring an increasingly large set of skills in different environments

D. a **virtual assistant** learning to help different users with different tasks at different points in time

E. a **doctor’s assistant** aiding in medical decision-making
<table>
<thead>
<tr>
<th>Desirable properties/considerations</th>
<th>Evaluation setup</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
What is the lifelong learning *problem statement*?

Problem variations:
- **task/data order**: i.i.d. vs. predictable vs. curriculum vs. adversarial
- **discrete** task boundaries vs. **continuous** shifts (vs. both)
- **known** task boundaries/shifts vs. **unknown**

Some considerations:
- model **performance**
- data **efficiency**
- **computational** resources
- **memory**
- others: privacy, interpretability, fairness, test time compute & memory

Substantial variety in problem statement!
What is the lifelong learning problem statement?

General [supervised] online learning problem:

for $t = 1, \ldots, n$
- observe $x_t$
- predict $\hat{y}_t$
- observe label $y_t$

**i.i.d. setting:** $x_t \sim p(x), y_t \sim p(y|x)$
- $p$ not a function of $t$

**otherwise:** $x_t \sim p_t(x), y_t \sim p_t(y|x)$

**streaming setting:** cannot store $(x_t, y_t)$
- lack of memory
- lack of computational resources
- privacy considerations
- want to study neural memory mechanisms

true in some cases, but not in many cases!
- recall: replay buffers
What do you want from your lifelong learning algorithm?

**minimal regret** (that grows slowly with $t$)

regret: cumulative loss of learner — cumulative loss of best learner in hindsight

$$\text{Regret}_T = \sum_{t=1}^{T} \mathcal{L}_t(\theta_t) - \min_{\theta} \sum_{t=1}^{T} \mathcal{L}_t(\theta)$$

(cannot be evaluated in practice, useful for analysis)

Regret that grows linearly in $t$ is trivial.  Why?
What do you want from your lifelong learning algorithm?

**Regret**: cumulative loss of learner — cumulative loss of best learner in hindsight

\[
\text{Regret}_T = \sum_{t=1}^{T} \mathcal{L}_t(\theta_t) - \min_\theta \sum_{t=1}^{T} \mathcal{L}_t(\theta)
\]
What do you want from your lifelong learning algorithm?

**positive & negative transfer**

**positive forward transfer**: previous tasks cause you to do better on future tasks compared to learning future tasks from scratch.

**positive backward transfer**: current tasks cause you to do better on previous tasks compared to learning past tasks from scratch.

positive -> negative : better -> worse
Plan for Today

The lifelong learning problem statement

Basic approaches to lifelong learning

Can we do better than the basics?

Revisiting the problem statement from the meta-learning perspective
Approaches

Store all the data you’ve seen so far, and train on it. → **follow the leader algorithm**

+ will achieve very **strong performance**
- **computation intensive** → **Continuous fine-tuning** can help.
- can be **memory intensive** [depends on the application]

Take a gradient step on the datapoint you observe. → **stochastic gradient descent**

+ **computationally cheap**
+ requires 0 memory
- subject to **negative backward transfer** “forgetting”
- slow learning

sometimes referred to as **catastrophic forgetting**
Very simple continual RL algorithm

Julian, Swanson, Sukhatme, Levine, Finn, Hausman, Never Stop Learning, 2020
Very simple continual RL algorithm

Julian, Swanson, Sukhatme, Levine, Finn, Hausman, Never Stop Learning, 2020
Very simple continual RL algorithm

What about negative transfer? Can we do better?

Julian, Swanson, Sukhatme, Levine, Finn, Hausman, Never Stop Learning, 2020
Plan for Today

The lifelong learning problem statement

Basic approaches to lifelong learning

Can we do better than the basics?

Revisiting the problem statement from the meta-learning perspective
Case Study: Can we modify vanilla SGD to avoid negative backward transfer? (from scratch)
Idea:
1. store small amount of data per task in memory
2. when making updates for new tasks, ensure that they don’t unlearn previous tasks

How do we accomplish (2)?

learning predictor $y_t = f_\theta(x_t, z_t)$ memory: $M_k$ for task $z_k$

For $t = 0, \ldots, T$

minimize $\mathcal{L}(f_\theta(\cdot, z_t), (x_t, y_t))$

subject to $\mathcal{L}(f_\theta, M_k) \leq \mathcal{L}(f_\theta^{t-1}, M_k)$ for all $z_k < z_t$

(i.e. s.t. loss on previous tasks doesn’t get worse)

Assume local linearity:

$\langle g_t, g_k \rangle = \langle \frac{\partial \mathcal{L}(f_\theta, (x_t, y_t))}{\partial \theta}, \frac{\partial \mathcal{L}(f_\theta, M_k)}{\partial \theta} \rangle \geq 0$ for all $z_k < z_t$

Can formulate & solve as a QP.

Lopez-Paz & Ranzato. Gradient Episodic Memory for Continual Learning. NeurIPS ‘17
Experiments

Problems:
- MNIST permutations
- MNIST rotations
- CIFAR-100 (5 new classes/task)

BWT: backward transfer,
FWT: forward transfer

Total memory size:
5012 examples

If we take a step back... do these experimental domains make sense?

Lopez-Paz & Ranzato. Gradient Episodic Memory for Continual Learning. NeurIPS ‘17
Can we meta-learn how to avoid negative backward transfer?


Beaulieu et al. *Learning to Continually Learn*. ‘20
Plan for Today

The lifelong learning problem statement

Basic approaches to lifelong learning

Can we do better than the basics?

Revisiting the problem statement from the meta-learning perspective
What might be wrong with the online learning formulation?

**Online Learning**  
(Hannan ’57, Zinkevich ’03)  
Perform sequence of tasks while minimizing static regret.

More realistically:

- **zero-shot performance**

- **slow learning**

- **rapid learning**
What might be wrong with the online learning formulation?

**Online Learning**  
(Hannan ’57, Zinkevich ’03)  
Perform sequence of tasks while minimizing static regret.

**Online Meta-Learning**  
Efficiently learn a sequence of tasks from a non-stationary distribution.

Primarily a difference in *evaluation*, rather than the *data stream*.

(Finn*, Rajeswaran*, Kakade, Levine ICML ’18)
The Online Meta-Learning Setting

for task $t = 1, ..., n$

observe $\mathcal{D}_t^{tr}$

use update procedure $\Phi(\theta_t, \mathcal{D}_t^{tr})$ to produce parameters $\phi_t$

observe $x_t$

predict $\hat{y}_t = f_{\phi_t}(x_t)$

observe label $y_t$

\textbf{Goal:} Learning algorithm with sub-linear

$$\text{Regret}_T := \sum_{t=1}^{T} \ell_t(\Phi_t(\theta_t)) - \min_{\theta \in \Theta} \sum_{t=1}^{T} \ell_t(\Phi_t(\theta))$$

(\text{Finn*}, \text{Rajeswaran*}, \text{Kakade}, \text{Levine} ICML '18)
Can we apply meta-learning in lifelong learning settings?

Recall the follow the leader (FTL) algorithm:
- Store all the data you’ve seen so far, and train on it.
- Deploy model on current task.

Follow the meta-leader (FTML) algorithm:
- Store all the data you’ve seen so far, and meta-train on it.
- Run update procedure on the current task.

What meta-learning algorithms are well-suited for FTML?

What if $p_t(T)$ is non-stationary?
Experiments

Experiment with **sequences of tasks**:
- Colored, rotated, scaled **MNIST**
- **3D object pose prediction**
- **CIFAR-100** classification

Example pose prediction tasks

plane

car

chair
Experiments

Comparisons:
- **TOE** (train on everything): train on all data so far
- **FTL** (follow the leader): train on all data so far, fine-tune on current task
- **From Scratch**: train from scratch on each task

**Follow The Meta-Leader**
learns each new task faster & with greater proficiency,
approaches **few-shot learning** regime
Takeaways

Many flavors of lifelong learning, all under the same name.

Defining the problem statement is often the hardest part

Meta-learning can be viewed as a slice of the lifelong learning problem.

A very open area of research.
Reminders

This Wednesday:  
Lecture and instructor office hours over zoom

Tuesday (Nov 30th):  
Project poster session

Wednesday (Dec 8th):  
Project due