Learning on the Job in the Open World

Bing Liu
Department of Computer Science,
University of Illinois at Chicago
Introduction

- Classic machine learning: **Isolated single-task learning**

  ![Diagram](image)

- **Key weaknesses**
  - Closed-world assumption: nothing new in testing / application
  - Model is fixed during application: no model revision/improvement in application
  - No knowledge accumulation: needs a large amount of labeled training data

- Suitable for well-defined tasks in restricted environments

- **This talk**: learning during application or on the job.

Closed-world assumption and open-world
(Fei et al, 2016; Shu et al., 2017)

- Traditional machine learning:
  - Training data: $D^{train}$ with class labels $Y^{train} = \{l_1, l_2, \ldots, l_t\}$.
  - Test data: $D^{test}, Y^{test} \in \{l_1, l_2, \ldots, l_t\}$

- **Closed-world**: $Y^{test} \subseteq Y^{train}$
  - Classes appeared in testing must have been seen in training, **nothing new**.
  - A system that is **unable to identify anything new**, it cannot learn by itself.

- **Open-world**: $Y^{test} - Y^{train} \neq \emptyset$
  - There are unseen classes in the test data
    - General case: out-of-distribution data
Chatbots need to learn continuously

- A chatbot’s environment is highly **dynamic & open**.
  - What happens if the user says something that the chatbot cannot understand.
  - Hard to train the chatbot by its engineers forever.
    - Chatbot must learn during chatting
    - (learn from other sources)
Self-driving cars need to learn continuously too

- Self-driving cars cannot reach human-level of driving with only rules and off-line training.
  - Impossible to cover all corner cases
  - Real-world is full of unknowns.
- Has to learn & adapt continuously in its interaction with humans and the environment by itself.
  - in the open world (changes & unknowns).

Outline

- Learning on the job continuously
- Detecting novel instances
- Continual learning with knowledge transfer
- Continuous learning in dialogues
- Summary
Old definition of lifelong learning

- At any point in time, the learner has learned a sequence of tasks, from 1 to $N$.

- When faced with the $(N+1)$th task, it uses the knowledge gained from the $N$ previous tasks to help learn the $(N+1)$th task.

Thrun. Is learning the n-th thing any easier than learning the first? NIPS, 1996.
Lifelong/continual learning in the open world

(Chen & Liu, 2018, Liu, 2020)
Key characteristics
(Chen and Liu, 2018-book)

- **Continuous learning process:**
  - *Without forgetting*: Learning a new task should not forget the past.
- **Knowledge accumulation in KB** (long-term memory)
- **Using/adapting past knowledge** to help learn new tasks
- **Learning on the job** during model application in the open world in a *self-supervised* manner via *interaction* with humans and the environment.
  - *Discover new tasks and learn them* incrementally/continually.
    - Novel instances of existing/known classes – concept drifting.
    - Novel/unknown classes or tasks
Learning on the job
(Liu, 2020, Chen and Liu, 2018)

- It is known in learning science that about **70% of our human knowledge comes from ‘on-the-job’ learning**.
  - Only about **10% through formal education or training**
  - About **70% from on-the-job learning**
  - The rest **20% through observation of others**

- **AI agents should also learn on the job or during applications**
  - the world is too complex and constantly changing.
  - Impossible to learn everything offline using manually labeled data

Learning on the job in the open-world
(Fei et al, 2016; Shu et al., 2017)

- **Steps:**
  - **Discover novel instances:** e.g., classify instances in $D^{\text{test}}$ to $Y^{\text{train}}$ and detect **novel instances** $D^{\text{novel}} \subseteq D^{\text{test}}$ belonging to $L_0$ – new tasks
  - **Identify the unseen/new classes** in $D^{\text{novel}}$, $L_0 = \{l_{t+1}, l_{t+2}, \ldots\}$ and gather training data
    - **Interactive self-supervision:** interaction with humans and the environment
  - **Continual learning:** Incrementally learn the new classes $\{l_{t+1}, l_{t+2}, \ldots\}$ (the new task)

Note: this does not include how the system should respond or react to novelty.
Interactive self-supervision

(Liu, 2020)

- Identify new classes and training data by interacting with
  - **Humans**: through natural language, e.g.,
    - Self-drive cars: asking the passenger
      - What is that object? How do I drive now? Where should I stop?
    - **Chatbots**: learn new knowledge and learn language during chatting.
  - **Environment**: get feedback & use tools (e.g., search engines)
    - Need an internal evaluation system
      - to evaluate environmental feedback

- **To gather knowledge, and supervisory or reward information.**
Example - a greeting bot in a hotel
(Chen and Liu 2018)

- See an existing/known guest.
  - Bot: “Hello John, how are you today?”

- See a new guest. Bot must recognize the guest is new/novel.
  - Bot: “Welcome to our hotel! What is your name, sir?”
  - Guest: “I am David”
  - Bot learns to recognize David automatically
    - take pictures of David and incrementally learn to recognize him

- See David next time.
  - Bot: “Hello David, how are you today?”
Outline

- Learning on the job continuously
- Detecting novel instances
- Continual learning with knowledge transfer
- Continuous learning in dialogues
- Summary
To detect novel instances that do not belong to training classes.

Figure 1: Overall Network of DOC

DOC - detecting novel instances (text classification)
(Shu et al. 2017)
Finding the rejection threshold

Figure 2: Open space risk of sigmoid function and desired decision boundary $d_i = T$ and probability threshold $t_i$. 
Open-world learning via meta-learning
(Xu et al. 2019)

L2AC–meta-learning

- It maintains a dynamic set $S$ of seen classes that allows new classes to be added or deleted without re-training.
  - Each class is represented by a small set of training examples.
- In testing, the meta-classifier uses only the examples of the seen classes on-the-fly for classification and rejection (novel)
Outline

- Learning on the job continuously
- Detecting novel instances
- **Continual learning with knowledge transfer**
- Continuous learning in dialogues
- Summary
Continual learning

- **Continual learning (CL)** learns a sequence of tasks.

- **CL has focused** on dealing with *catastrophic forgetting*
  - **Catastrophic forgetting:** In learning a new task, the learner should not forget what it has learned from previous tasks.
  - Extensive research has been done

- **CL should also leverage** the knowledge learned from previous tasks to learn the new task better.
Continual learning with knowledge transfer
(Ke and Liu, 2020)

- Task-based continual learning (TCL)
  - Each task is an independent classification problem.

- Proposed system: KAN (knowledge accessibility network)

- Application: continual sentiment classification
  - Goal: learn a sequence of sentiment classification tasks.
  - Each task – classify reviews of a category of products.
    - Review: “I bought a cellphone a few days ago. It is such a nice phone. The touch screen is really cool. The voice quality is great too. ....”
    - Class labels: positive or negative
KAN architecture

- **Accessibility (AC) module**
  - decides accessible units in the KB by the current task \( t \) by learning a binary mask \( a_t \).

- **Main continual learning (MCL)**
  - **Knowledge base** (KB-RNN).
  - performs the main continual learning and testing.
  - Uses mask \( a_t \) to block not-useful units in KB (avoid forgetting)
    - Useful units transfer knowledge.
Experiment results

- 24 sentiment classification tasks & 7 baselines

Average accuracy

<table>
<thead>
<tr>
<th>Models</th>
<th>All Tasks</th>
<th>Last Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>ONE</td>
<td>0.7846</td>
<td>0.7809</td>
</tr>
<tr>
<td>LSC</td>
<td>0.8219</td>
<td>0.8246</td>
</tr>
<tr>
<td>N-CL</td>
<td>0.8339</td>
<td>0.8477</td>
</tr>
<tr>
<td>EWC</td>
<td>0.6899</td>
<td>0.7187</td>
</tr>
<tr>
<td>OWM</td>
<td>0.6983</td>
<td>0.7337</td>
</tr>
<tr>
<td>HAT</td>
<td>0.6456</td>
<td>0.6938</td>
</tr>
<tr>
<td>SRK</td>
<td>0.8282</td>
<td>0.85</td>
</tr>
<tr>
<td>KAN</td>
<td>0.8524</td>
<td>0.8799</td>
</tr>
</tbody>
</table>

Effect of forward and backward transfer

<table>
<thead>
<tr>
<th>Tasks</th>
<th>ONE</th>
<th>N-CL Forward</th>
<th>N-CL Backward</th>
<th>KAN Forward</th>
<th>KAN Backward</th>
</tr>
</thead>
<tbody>
<tr>
<td>First 6 tasks</td>
<td>0.7846</td>
<td>0.7937</td>
<td>0.7990</td>
<td>0.8068</td>
<td>0.8132</td>
</tr>
<tr>
<td>First 12 tasks</td>
<td>0.7865</td>
<td>0.8135</td>
<td>0.8199</td>
<td>0.8314</td>
<td>0.8390</td>
</tr>
<tr>
<td>First 18 tasks</td>
<td>0.7870</td>
<td>0.8253</td>
<td>0.8327</td>
<td>0.8424</td>
<td>0.8501</td>
</tr>
<tr>
<td>First 24 tasks</td>
<td>0.7846</td>
<td>0.8302</td>
<td>0.8339</td>
<td>0.8471</td>
<td>0.8524</td>
</tr>
</tbody>
</table>
Outline

- Learning on the job continuously
- Detecting novel instances
- Continual learning with knowledge transfer
- Continuous learning in dialogues
- Summary
Continuous knowledge learning in dialogues
(Mazumder et al. 2018, 2019)

- Dialogue systems are increasingly using knowledge bases (KBs) storing real-world facts to help generate responses.
  - KBs are inherently incomplete and remain fixed,
  - which limit dialogue systems’ conversation capability

- **CILK**: *Continuous and Interactive Learning of Knowledge* for dialogue systems
  - to continuously and interactively learn and infer new knowledge during conversations
Human conversation is knowledge driven?

Knowledge grounding makes conversation **interesting** and **intelligent**.
Knowledge learning in conversation

Humans Learn and Leverage Knowledge in Lifelong Manner!

Hey, I visited Stockholm last week. The place is awesome!

Where is Stockholm?

Stockholm is the capital of Sweden

Hey, I am planning for a Europe tour soon

Are you visiting Stockholm? I heard the place has lot of attractions

Knowledge learning happens in a multi-user environment
Opportunities to learn in conversations

1. Extracting knowledge directly from user utterances. E.g.,
   - User: Obama was born in Hawaii.
   - Agent extracts: (Obama, BornIn, Hawaii) – expressed in triples \((h, r, t)\)

2. Asking user questions & expecting correct answers, e.g.,
   - Agent: Where was Obama born?
   - User: Hawaii => (Obama, BornIn, Hawaii)

3. **When the agent cannot answer user questions**, it asks the user for some supporting facts and then infers the answers.
   - We focus on this setting (which covers 1 and 2)
Problem formulation

- Given a user query / question \((h, r, ?)\) [or \((?, r, t)\)], our goal is two-fold:
  1. **Answering** the user query or **rejecting** the query to remain unanswered if the correct answer is believed to not exist in the KB
  2. **learning / acquiring** some knowledge (supporting facts) from the user to help the answering task.

- We further distinguish two types of queries:
  1. **Closed-world Queries**: \(h\) (or \(t\)) and \(r\) are **known** to the KB
  2. **Open-world Queries**: Either one or both \(h\) (or \(t\)) and \(r\) are **unknown**

→ Proposed Soln.

an engine for Continuous and Interactive Learning of Knowledge (CILK)
Interactive knowledge learning in dialogue: example

(Mazumder et al. 2019)

USER: In what Country is Boston located?

CILK: I do not know what “located in Country” means? Can you provide me an example?

USER: (London, LocatedInCountry, UK). “London is located in UK.”

CILK: Got it. Can you tell me a fact about “Boston”? 

USER: (Harvard University, UniversityLocatedIn, Boston). “Harvard university is located in Boston.”

CILK: (Boston, LocatedInCountry, USA) “Boston is located in USA.”

Mazumder, Liu, Wang, and Ma. Lifelong and Interactive Learning of Factual Knowledge in Dialogues. SIGDIAL-2019
Components of the system

- **Knowledge Base** of triples
- **Interaction module**: chatting with the user
  - decides whether to ask or not and formulates questions to ask the user for supporting facts
  - Updates KB with the acquired knowledge (used for training)
  - Converts open-world queries to closed-world ones
    - Using support facts to make h (or t) and r known to the KB (added to KB).
- **Inference module**: using the acquired knowledge (supporting facts) and KB to answer the resulting closed-world query: (h, r, t).
Natural language interface and grounding

**Motivation**: when we were testing a self-driving car on the road, it suddenly stopped and refused to move.

- The road was completely clear and we could not see anything wrong
  - Debugging back in the lab found a small stone on the road.
- Why cannot the car tell us what the problem was in natural language (NL)?
- Why cannot we tell the car to go ahead in natural language?
  - Our instruction is a piece of supervisory information.

**To have this capability, the agent needs to continuously learn**

- new natural language expressions that it does not understand, and
- to ground them to appropriate actions


Continual Learning workshop @ ICML-2020, July 17, 2020
Outline

- Learning on the job continuously
- Detecting novel instances
- Continual learning with knowledge transfer
- Continuous learning in dialogues
- Summary
Classic ML: isolated, closed-world, single-task learning

AI agents must learn continuously in the open world, i.e.,
- Learning on the job (Chen and Liu, 2018; Liu, 2020)
  - Detect new things and learn them with
    - self-motivation & self-supervision
    - Interactive self-supervision: need to interact with humans and the environment
- E.g., self-driving cars and chatbots need such capabilities.

Current techniques are still in their infancy.
- Novel learning algorithms or paradigm shifts may be needed.
Thank You

Q&A