Achieving Forgetting Prevention and Knowledge Transfer in Continual Learning

Zixuan Ke[1], Bing Liu[1], Nianzu Ma[1], Hu Xu[2] and Lei Shu[3]
University of Illinois at Chicago[1], Facebook AI Research[2] and Amazon AWS AI[3]

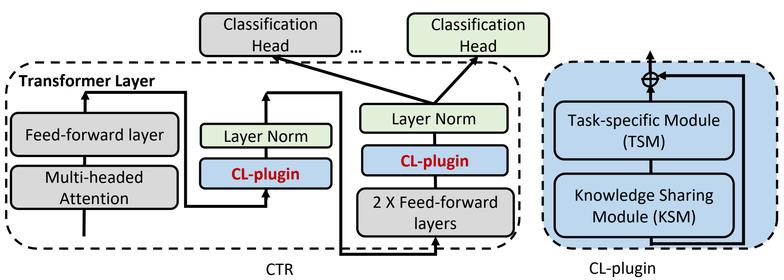
NEURAL INFORMATION PROCESSING SYSTEMS

Capsule and Transfer Routing for Continual Learning

- Continual learning learns a sequence of tasks
  - **CTR** focuses on task incremental learning (Task-CL), where task id is known in training and testing
- Existing research mainly focused on **Overcoming Catastrophic Forgetting**
  - Assume that tasks are dissimilar and have little shared knowledge
- Some works also addressed **Knowledge Transferring**
  - Leverage the past knowledge to help learn the new task when tasks are similar and have shared knowledge
  - Works well for similar tasks, but may have serious forgetting for dissimilar tasks
- **Our goal:** Achieve both **Forgetting Prevention** and **Knowledge Transfer** for both **similar** and **dissimilar** tasks.

Proposed CTR Model: CL-Plugin

- In NLP, fine-tuning a BERT-like pre-trained model has become a standard
- In CL, **however**, most techniques do not use pre-trained models
- How to make the best use of pre-trained models in CL

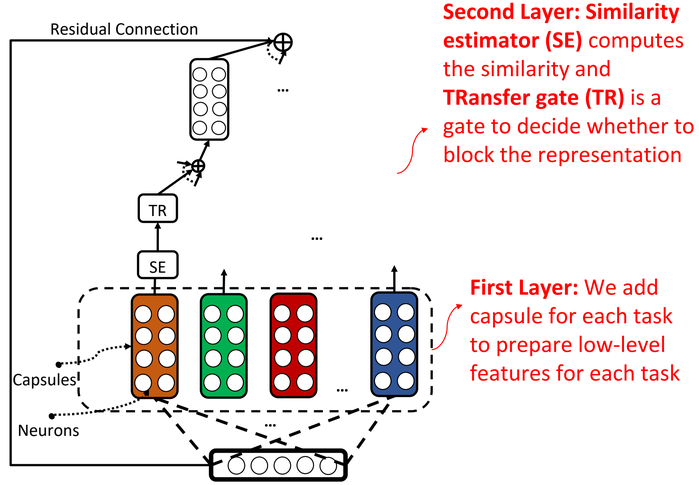


- **CTR** inserts a continual learning plugin **(CL-plugin)** in two locations in BERT's transformer layers, inspired by Bert adapter (Houlsby et al., 2019).
- We no longer need to fine-tune BERT for each task, which causes CF in BERT, and yet we can achieve the power of BERT fine-tuning

Proposed CTR Model: Overcome Forgetting

- **CTR** protects previous knowledge by the **task mask mechanism**
  - Task ID *t* is an input in training, and is used to train neurons masks, indicating which neurons are for which task
  - The task masks are **stored** after training of each task so that
    - Used neurons by a task can be protected by applying its masks during the training of subsequence tasks
    - Knowledge of a previous task can be obtained by opening its corresponding task masks

Proposed CTR Model: Selectively Knowledge Transfer

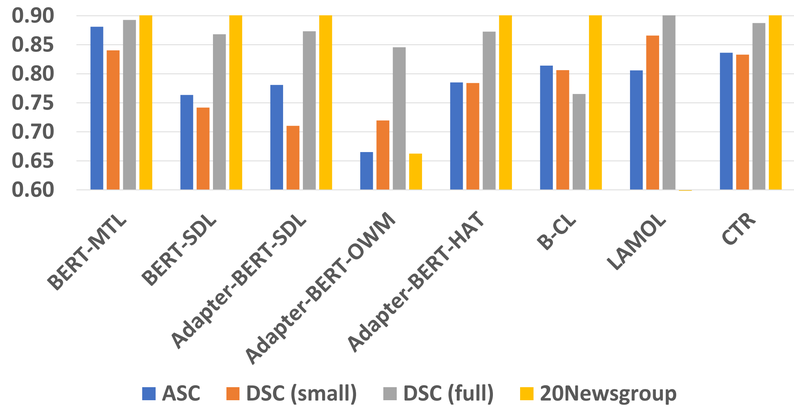


Second Layer: Similarity estimator (SE) computes the similarity and **TRansfer gate (TR)** is a gate to decide whether to block the representation

First Layer: We add capsule for each task to prepare low-level features for each task

- **Knowledge transfer module**, which is a Capsule Network with the proposed Transfer Routing
  - **First layer** (encode additional information for each task)
    - Add a capsule for each task and prepare low-level features derived from each task
  - **Second layer** (compute similarity and block dissimilar tasks)
    - Compute the **similarity** between previous task capsules and current task capsules per data instance
    - **Based on the similarity**, we train a gate function to decide whether to block the task capsules and then aggregates the features of unblocked task capsules to obtain a good task-shared representation.
    - **As a result,** the capsule network can block dissimilar task capsules (by setting their gates to 0) and open similar task capsules (by setting their gates to 1) during training, which encourages positive knowledge transfer.

Experimental Results



- **CTR** outperforms ALL 37 + considered baselines
  - Yields the best Acc. and MF1 for similar tasks (except LAMOL in DSC (full). Note that LAMOL is based on GPT-2 backbone)
  - Yields least forgetting for dissimilar tasks (LAMOL has sever forgetting in 20Newsgroup, while B-CL and CTR have much less)
  - Results similar to those of MTL, which is regarded as the upper bound of continual learning

Summary

- ✓ We studied
  - How to achieve both forgetting prevention and knowledge transfer in task continual learning (Task-CL) setting
- ✓ We proposed **CTR:** a novel CL-plugin is inserted in BERT to improve fine-tuning BERT performance on the continual learning setting
  - **Overcome Forgetting:** task mask mechanism is used to protect knowledge for each previous task
  - **Knowledge transfer: Capsule network** with **transfer routing** are used to extract transferrable knowledge.
- ✓ Experimental results show that
  - CTR markedly improves the similar and dissimilar tasks performance

Code and data (and 20+ baselines!): https://github.com/ZixuanKe/PyContinual