Robust multi-label learning for weakly labeled data

Author: Atanas Marinov
Responsible professor: Lydia Chen
Supervisors: Amirmasoud Ghiassi, Taraneh Younesian

Background information

- **Multi-label learning** (MLL) is a type of supervised learning in which each input example could be assigned to more than output label.
- **Weak labels** are a type of multi-label data corruption where not only all the relevant (true) labels are presented but also some of the irrelevant ones.

Example image

Weak labels
Bear
Bike
Bed
Backpack
Book

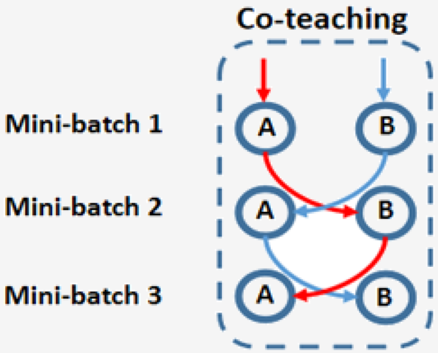
Correct labels
Bear
Backpack
Book

Overfitting problem - The deep neural networks used to solve MLL problems could be quite complex and often have a huge capacity. This enormous capacity, however, could also be a negative, as they tend to eventually **overfit** the undesirable corrupted labels.

Method: Co-ASL

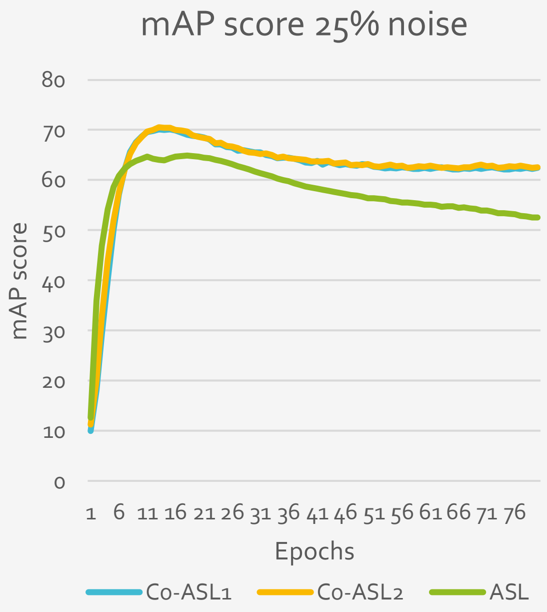
Co-ASL solves the problem by combining the most prominent solutions in the two fields - **ASL**[1], the state-of-the-art MLL approach, and **Co-teaching**[2], an eminent robust training strategy.

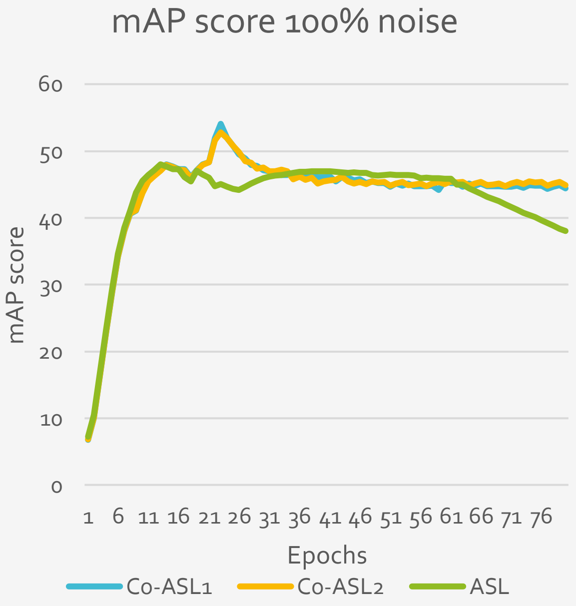
- Initialize the two peer networks;
- The data is fed in a mini-batch manner;
- Each network forwards the entire mini-batch and remembers the portion with the lowest loss (small-loss instance);
- The networks swap their small-loss instances and adjust their weights using only the instance of their peer.



Results

- **MS-COCO**[3] was used as a dataset for the experiments.
- It is a clean dataset and, therefore, an **artificial noise** was injected into it.
- The experiments were running for **80 epochs**.





Conclusions

Co-ASL proves to be an MLL algorithm robust for weakly labeled data.

- **Does not overfit** even after 80 epochs.
- Achieves an improvement of **8-9 mAP** score over ASL.

CSE3000

TU Delft

[1] Emanuel Ben-Baruch, Tal Ridnik. *Asymmetric loss for multi-label classification*
[2] Bo Han, Quanming Yao. *Coteaching: Robust training of deep neural networks*
[3] Tsung-Yi Lin, Michael Maire. *Microsoft COCO: common objects in context*