Handin 3 - Semi-supervised deep learning

Rasmus Freund - 201700273

November 28, 2023

Short answer questions

- Where are we ensuring that the finetuning does not affect the feature extractor?
  - The `Net` class has been created with separate components for feature extraction and classification; which has additionally been split into two layers: `pretrainer` and `generalizer`. When finetuning (`optim.SGD(network.generalizer.parameters(),lr=0.01)`) we only update the `generalizer` parameters, not those of the feature extractor.
- How dose the code work that gets s samples per class for the pretraining dataset
  - The `get_subsampled_dataset` function iterates over each class in the dataset. For each class, it finds the indices where the target label matches the current class and randomly selects k indices from that list which are then appended to the subset.
- What is the `forward_call` parameter responsible for in the `train()` method (located in `network_training.py`)?
  - It performs a forward pass through the network, calculating the loss.
- Describe how the `augment()` method works (located `augmentations.py`).
  -
- If we pre-train and finetune on the same dataset, is there any reason to do the finetuning step?
  -

Predictions

- Will the collage and mixup data augmentations help achieve higher finetune accuracies? Which do you expect will be more effective?
  -
- What relationship do you expect between the number of samples in the pre-training dataset and the finetuning accuracy? Does this change with data augmentations?
  -