

Weekly Report KARIMI 2021-08-27

My work this week has mainly been towards

1. Arxiv papers + Paddle Paddle
2. Federated and Distributed EM paper
3. AAAI22 Submission
4. Compression and Generative Modeling

1 Arxiv papers + Paddle Paddle

Worked on the Fed-LAMB and Spars-AMS codebase in paddle paddle. And the arxiv paper fro Fed-LAMB.

Arxiv for Spars-AMS will be also a priority once we finish to **add 1-bit Adam paper (ICML21) as a baseline and add new experiment on imagenet or tiny imagenet.**

2 Federated EM paper

Added an experiment in the paper.

Most of my progress is on overleaf. The algorithms are written as their final versions (because they work in practice).

However, the theoretical assumptions are not final since I have not finished the theory part yet.

3 STANLEY paper

Final touches for AAAI22 submission.

Currently debugging a problem on celeba dataset in order to improve the current results.

4 Compression and Distributed Generative Models

Compression for memory saving: At the sampling step, there might be a need for more memory to store a lot of negative samples. If this pain point is verified, then compression can be used in the MCMC to sample more negative samples while using less memory storage. The result is that the approximation of the gradient will hence be more accurate (because we use more samples) without increasing the memory storage.

Compression for distributed computing: Xiaoyun raised another point which is that if EBM has a need for distributed computing (and maybe federated), then compression can be used to alleviate the low bandwidth of the workers. This could be a project but I have to verify that there is a need for distributing the EBM training.

The most urgent question to answer for this is:

- How to construct a Markov Chain with compressed states (the idea is to obtain compressed negative samples and hence never store any high resolution chain states)
- Once chains are built with only compressed states, then the user will be able to increase drastically the number of chains to run in parallel (no more memory issue) and hence have a larger MC batch, of compressed images
- The approximation of the gradient is hence better. Yet gradient vectors need to be computed, which can occupy a lot of memory. We have to think about this crucial point.