**Sparsity-Preserving Differentially Private Training of Large Embedding Models**

**Introduction**

Results

**Embedding Models Process (Private) Non-numerical Inputs**

1. Non-numerical data
   - Demographic data
   - Private chats
   - Medical history

2. Embeddings

   - Sparse lookup → Sparse gradients (leveraged by customized APIs such as Google TPUs for efficiency)

3. Downstream tasks
   - Recommender systems
   - Chatbots
   - ML-powered diagnosis

**Recommendation Tasks**

- Sparse gradients → Dense gradients (more computation)

**Natural Language Understanding Tasks**
- SQuAD, QQP from GLUE benchmark

**Comparison w/ LoRA [3]**

- DP-AdaFEST achieves sparser gradients compared to LoRA, which adapts weight matrices using low-rank approximation
- DP-AdaFEST benefits from the efficient embedding lookup via customized APIs. LoRA would not be able to leverage them (it requires relatively expensive matrix multiplication)

**References**

2. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. Wang et al., ICLR 2019
3. LoRA: Low-rank Adaptation of Large Language Models. Hu et al., ICLR 2022
5. Unsupervised Cross-lingual Representation Learning at Scale. Conneau et al., ACL 2020

**Conclusion**

**Future Work**

- Leverage specialized hardware to further optimize the computational performance and speed up the training

- Integration of our methods with non-centralized training paradigms (e.g., Federated Learning)

**Method**


- We extend standard DP-SGD with an extra mechanism at each iteration to privately select the "top features":
  1. Compute how many examples contributed to each non-numerical feature "bucket"
  2. Restrict the total contribution from each example by clipping their counts
  3. Add Gaussian noise to the contribution count of each feature bucket
  4. Select only the features to be included in the gradient update that have a count above a given threshold (a sparsity-controlling parameter) to be included in the gradient update, thus maintaining sparsity.

**Takeaways**

- We effectively address the "destroyed gradient sparsity" challenge when applying general-purpose DP-SGD to large-scale embedding models, via the proposal of DP-AdaFEST
- DP-AdaFEST achieves a substantially sparser gradient in recommendation tasks, with a reduction in gradient size of over $10^5$ (translates into 20x wall-clock time improvement) compared to the dense gradient produced by vanilla DP-SGD, while maintaining comparable levels of accuracy.
- DP-AdaFEST is also more effective than LoRA in reducing the gradient size when applied to natural language understanding tasks.

**Results**

**Recommendation Tasks**

- Criteo-Kaggle & Criteo-1T (Time-series). Vocab size: 7.6M.
- > 10x% reduction in gradient size w/ comparable utility
- > 20x wall-clock-time improvement in simulation

**Natural Language Understanding Tasks**

- SQuAD, QQP from GLUE benchmark
- Vocabulary size: ~50k
- ~50x reduction in gradient size w/ comparable utility (due to already condensed vocabulary)

**Conclusion**

**References**

2. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. Wang et al., ICLR 2019
3. LoRA: Low-rank Adaptation of Large Language Models. Hu et al., ICLR 2022
5. Unsupervised Cross-lingual Representation Learning at Scale. Conneau et al., ACL 2020