Capstone Bi-Monthly Update

Date: 10/25/2023

Team: Aditya, Elisa, Jenny, Zhanyi

Duke

Agenda

1) Network Acceleration Experiment Discussion
    a) Updates & Results
    b) Challenges
2) Next Steps

Duke

Network Acceleration Experiment

1) Last Time:
    a) Performed experiments on Distilled and Quantized BERT models
        i) Issues with the experiments:
            1) Done On Different Datasets
            2) Done On Different Machines

2) Today:
    a) Unified our experiments and performed pruning on BERT model
    b) Formalized Experiment Scope (Would love to get your feedback on it)

Duke

BERT - Network Acceleration Experiment - Results

Method	Accuracy	Training Time (s)	Inference Time (s)	Model Size (MB)
Industry Benchmark	~39%*	N/A	N/A	N/A
BERT - (FineTuned) Baseline	28.10%	384.89	0.05 +- 0.02	417.68
BERT - Pruning	26.07%	393.84	0.012 +- 0.0008	417.68
BERT - Distillation	27.09%	465.88	0.02 +- 0.01	255.44
BERT - Quantized	28.55%	N/A	0.03 +- 0.01	91.08
BERT Distillation + Quantized	25.05%	N/A	0.02 +- 0.01	91
BERT Distillation + Pruning				
BERT Pruning + Quantized	23.36%	N/A	0.03 +- 0.01	91.08
BERT Distillation + Pruning + Quantized				

Machine: GPU (T4)

Task: Multi-Label Classification (Num of Labels: 11)

Dataset: Emotion Classification on Tweets (Num of Records: ~11k)

# of Epochs: 5

Train-Val-Test Split: 60%-30%-10%

* Not necessarily trained using BERT Model.

Information obtained from 2018 Data Science CodaLab competition

Duke

Network Acceleration Experiment - Challenges

1. Quantization (pytorch)
   a. Does not gel with high-level huggingface abstractions
   b. Works best with pytorch code
2. Distillation
   a. Noted that finding a suitable student architecture is challenging and a emerging research area in the LLM space
   b. However, we believe it is still possible to apply distillation to other acceleration methods (quantized and pruned models) that could potentially yield promising results
3. Pruning
   a. Fine-tuning Requirement
   b. Determining the Optimal Pruning Ratio and Pruning Strategy

Network Exchange Experiment & Challenges

1. Results:
   - Applied ONNX on distilled BERT
   - Got similar inference time on simple Task

Model	Inference Time (seconds)
ONNX Model	0.31246399879455566
PyTorch Model	0.34558868408203125

2. Challenges:
   - RAM limitation causes crush
   - Runtime is not obvious faster than original framework

Duke

Next Steps

1. Expand to other LLMs and 2-3 larger additional datasets to test the generalization of results.
2. Cont. literature review on best practices for each acceleration method
3. Verify ONNX model and solve memory issues
4. Add optimizing features of ONNX model (such as constant folding and runtime optimization)