ANLP

Assignment-4

Report

Akshit Sharma
(2021101029)

Model Used: GPT-2 (small, 117 million parameters)

Base model (no quantization, 32-bit precision)

Model size: **475 MB**

Inference latency (Average of 3000 samples): **0.08412 s**

Perplexity (Average of 3000 samples): **29.28**

PART-1: Quantization from Scratch

Full Model Quantization (to INT8)

Model size (post quantization): **119 MB**

Inference latency (Average of 3000 samples, including time for dequantization): **0.12026 s**

Perplexity (Average of 3000 samples): **34.86**

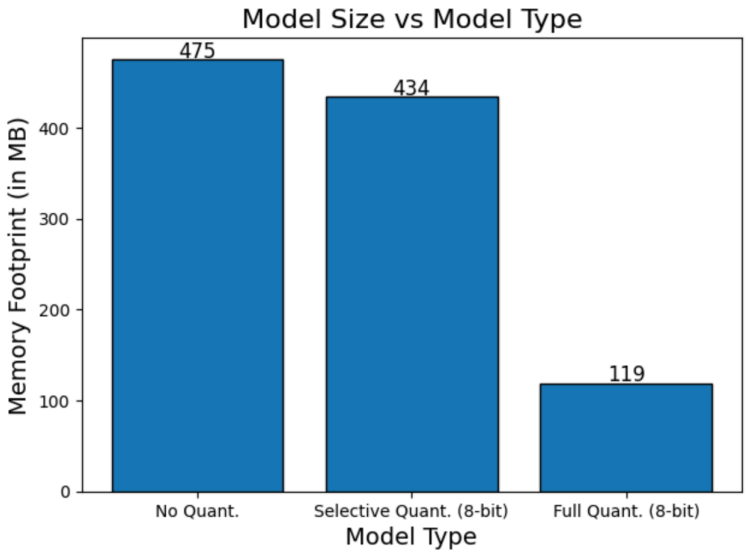
Selective Component Quantization (to INT8)

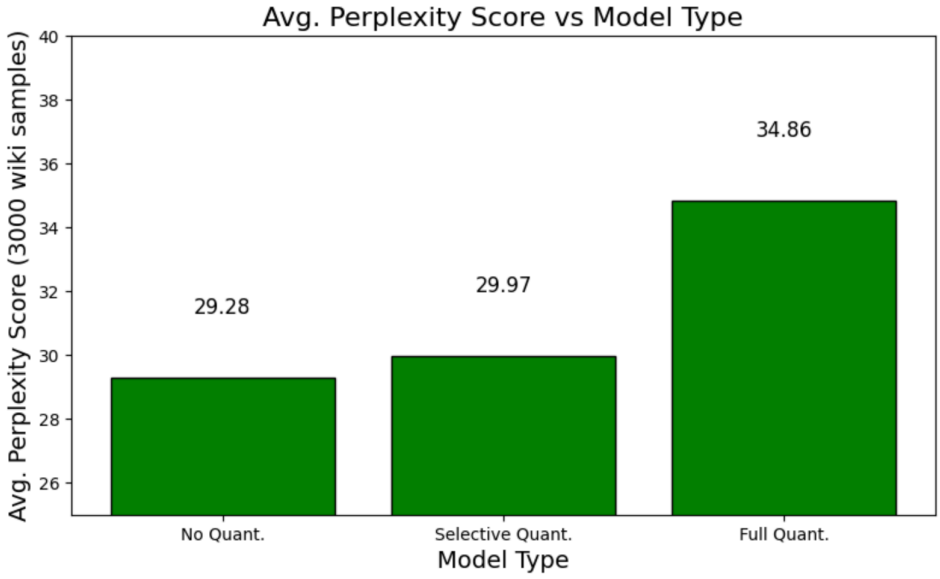
Components Quantized: **2 blocks** of decoder- **model.transformer.h[0], model.transformer.h[1]**

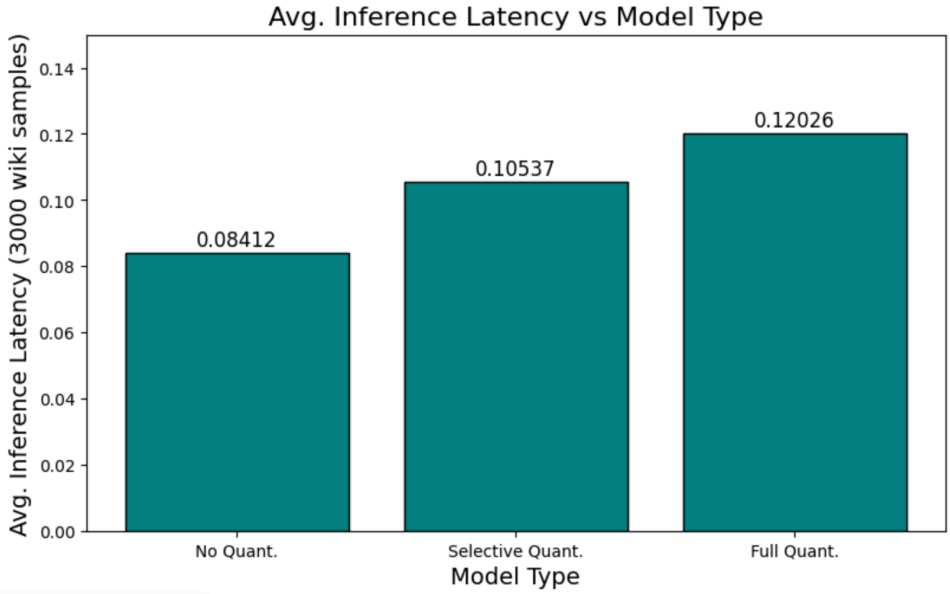
Model size (post quantization): **434 MB**

Inference latency (Average of 3000 samples, includes time for dequantization): **0.10537 s**

Perplexity (Average of 3000 samples): **29.97**


Model Size vs Model Type
Memory Footprint (in MB)
0
100
200
300
400
475
434
119
No Quant.
Selective Quant. (8-bit)
Full Quant. (8-bit)
Model Type


Avg. Perplexity Score vs Model Type
Avg. Perplexity Score (3000 wiki samples)
26
28
30
32
34
36
38
40
29.28
29.97
34.86
No Quant.
Selective Quant.
Full Quant.
Model Type


Avg. Inference Latency vs Model Type
Avg. Inference Latency (3000 wiki samples)
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.08412
0.10537
0.12026
No Quant.
Selective Quant.
Full Quant.
Model Type

Analysis of Observations

- **Perplexity Score:** Quantizing all layers reduces model precision, increasing perplexity. Quantizing only parts of the model retains the precision of most layers, which leads to a smaller increase in perplexity compared to full quantization.
- **Inference Latency:** Since these times include the de-quantization time as well, the average inference latency is higher for fully quantized model because all the model weights have to be de-quantized. For selective quantization, less time is taken for de-quantization as fewer weights are converted to INT8 precision.
- **Model Size:** All layers are converted to lower precision i.e. INT8, leading to a significant reduction in size - around 4x when moving from float32 to INT8. Size reduction is proportional to the fraction of the model quantized. So, the size of selectively quantized model is larger.

PART-2: Bitsandbytes Integration and NF4 Quantization Theory Questions

Concept of NF4 quantization and how it differs from linear quantization scales

NF4 quantization, or NormalFloat4 quantization, represents weights using 4 bits with a special logarithmic-like scaling. Instead of linearly mapping values to the quantized range, it focuses on maintaining more precision for smaller magnitudes, which are often more critical in neural networks.

Key Features:

- **Logarithmic-like Scaling:** NF4 uses a predefined lookup table of 16 float values, distributed non-linearly, to approximate the original weights. This non-linearity ensures better precision for small weight values.
- **Efficient Weight Representation:** By focusing on important weight ranges, NF4 minimizes the degradation of performance during quantization.

Comparison with Linear Quantization

Scaling Method

- **Linear Quantization:** Maps floating-point values to integer ranges using a uniform scale and zero point, maintaining equal step sizes.
- **NF4 Quantization:** Uses non-linear, pre-defined scale values, prioritizing smaller magnitudes.

Precision

- **Linear Quantization:** Precision is consistent across the range but struggles with representing small magnitudes accurately.
- **NF4 Quantization:** Offers finer granularity for smaller values, which are more sensitive in neural networks.

Performance

NF4 can yield better model performance at the same bit-width due to its focus on preserving crucial small weights, especially in sparse and pre-trained models.

Bitsandbytes Quantization

Note: Double quantization not used, GPU used for inference

8-bit precision

Model size: **230 MB**

Inference latency (Average of 3000 samples): **0.11998 s**

Perplexity (Average of 3000 samples): **29.80**

4-bit precision (FP4)

Model size: **195 MB**

Inference latency (Average of 3000 samples): **0.09347 s**

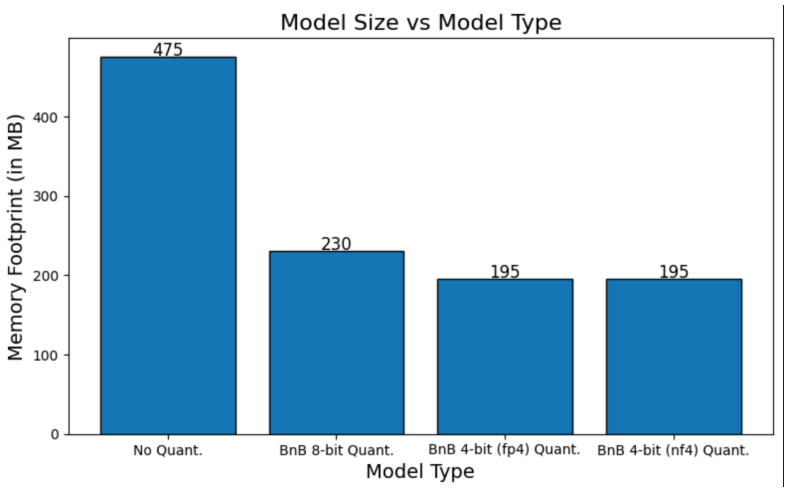
Perplexity (Average of 3000 samples): **32.55**

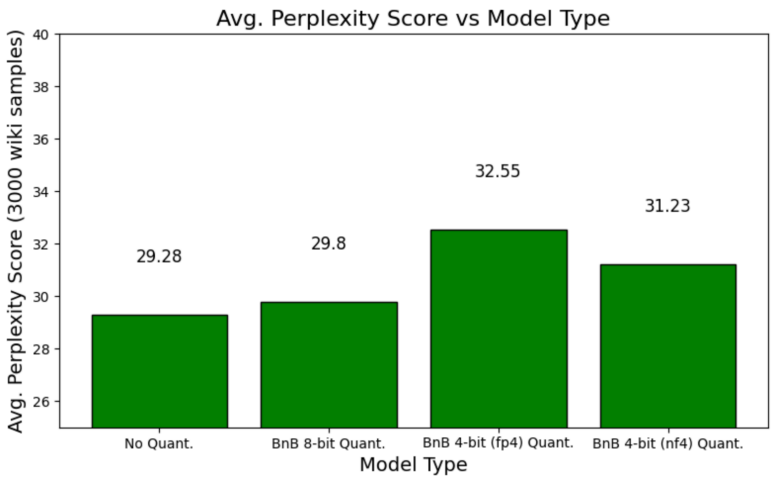
NF4 Quantization

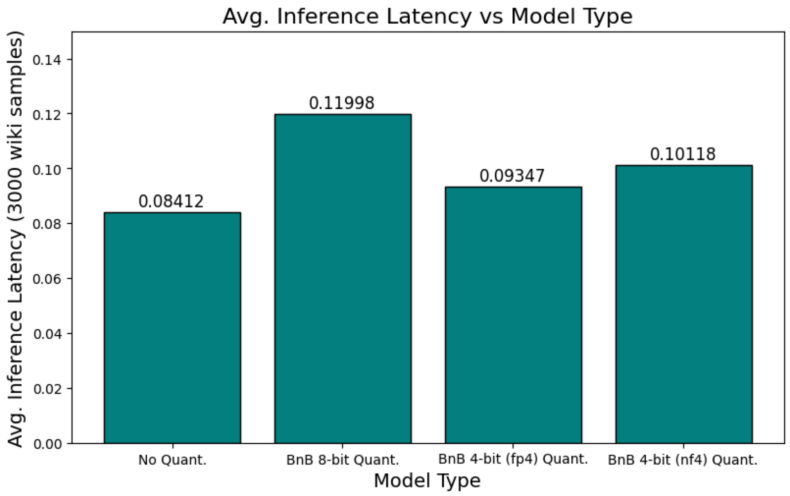
Model size: **195 MB**

Inference latency (Average of 3000 samples): **0.10118 s**

Perplexity (Average of 3000 samples): **31.23**







Comparing FP4 and NF4

Perplexity: We observe a higher average perplexity score for FP4, compared to NF4. This is because linear quantization distributes precision uniformly, which can lead to more significant degradation of smaller weights. NF4 has lower average perplexity due to its focus on preserving the precision of small values. This aligns better with the importance of small magnitudes in pre-trained models.

Inference Time: FP4 has slightly lower average inference latency compared to NF4, as linear quantization involves simpler arithmetic operations (multiplication and addition) during de-quantization. NF4's need for table lookups and non-linear scaling computations during weight reconstruction increases its inference latency.

Model Size: Same, as both convert the weights to 4-bit precision.

Conclusion

We observe that quantization takes a toll on model performance, visible in the higher average **perplexity** scores of quantized models compared to non-quantized model, because of loss of precision. **Model size** significantly reduces post quantization which makes the technique extremely useful for deploying LLMs on edge devices and in computationally constrained environments. **Inference time** is slightly higher for quantized models because time for de-quantization is added.