Quantization of large ML models

Introduction

Any approximation means a trade-off:

- Predictive Accuracy (Degree of Precision)
- Predictive Efficiency (Compute and Memory consumption)

Questions:

- When does the trade-off make sense?
- Which quantization level is optimal?
- What are the implications for deployment?

Part 1: Object Detection Models

Experiment:

- Model: MobileNetV2
- Dataset: COCO 2017
- Evaluation: Inference Time vs. IoU Precision
- Quantization: `float32`, `float16`, `int8`

Part 1: Object Detection Models

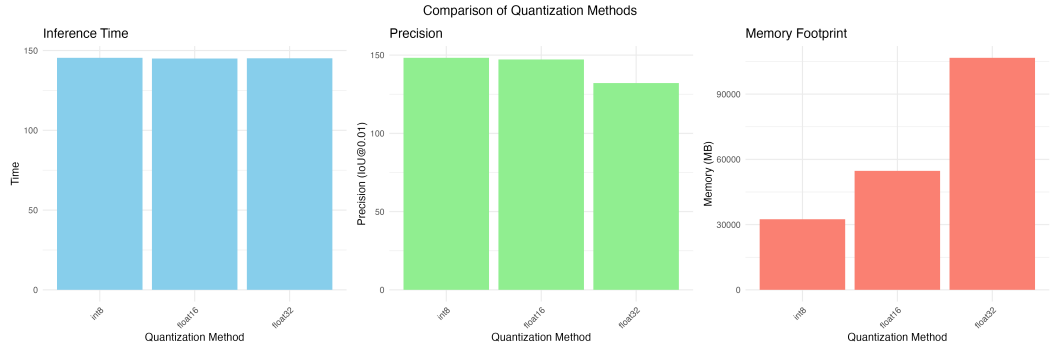
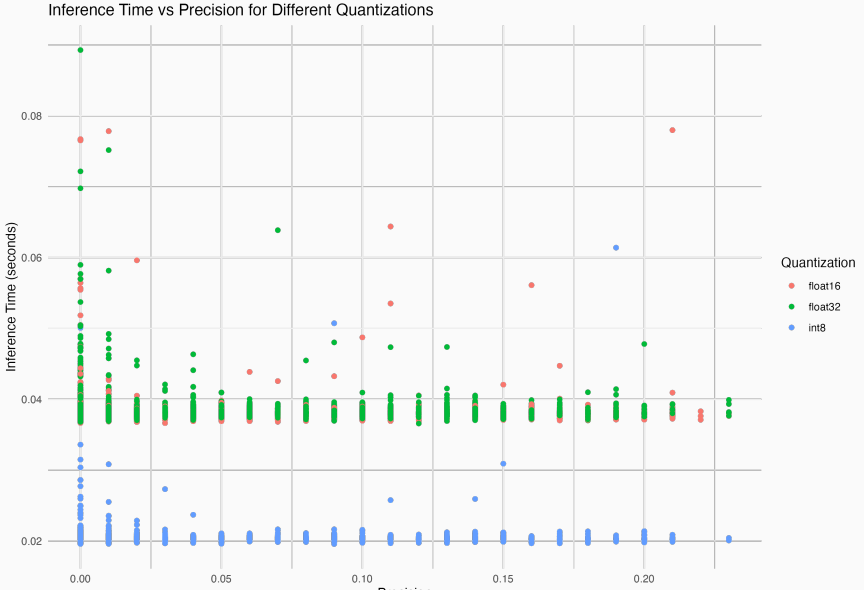
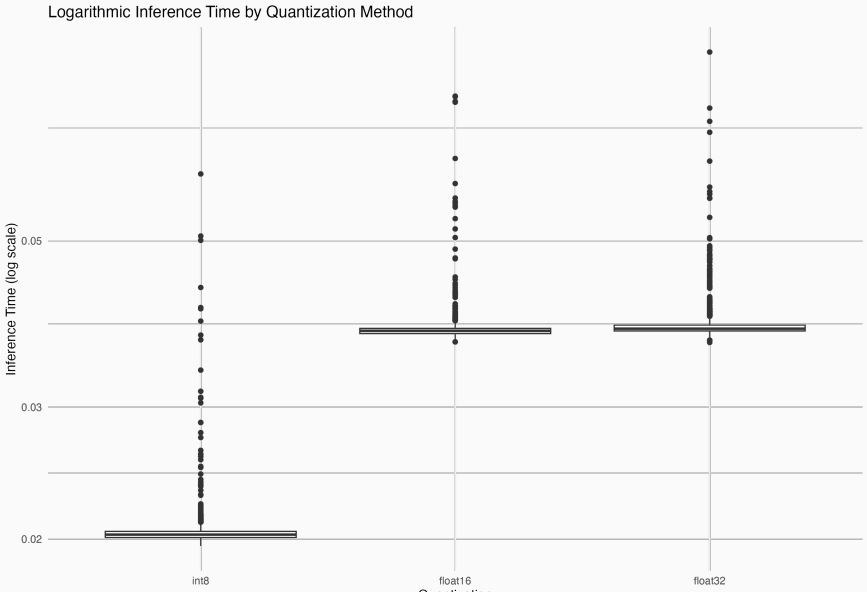


Figure 1: ODM: Inference Time vs. IoU Precision Bar plots

Part 1: Object Detection Models



Part 1: Object Detection Models



Conclusion: The results are inconclusive.

Further inspection to is necessary to validate our findings.

If these findings were to be correct, the int8 quantized model would be the best choice for any deployment scenario as it is the most efficient in terms of memory, inference speed and precision.

Experiment:

- Model: SmolLM-135M
- Dataset: LAMBADA
- Evaluation: Top-1 and Top-5 Accuracy
- Quantization: `int8`, `int4`, `int2`

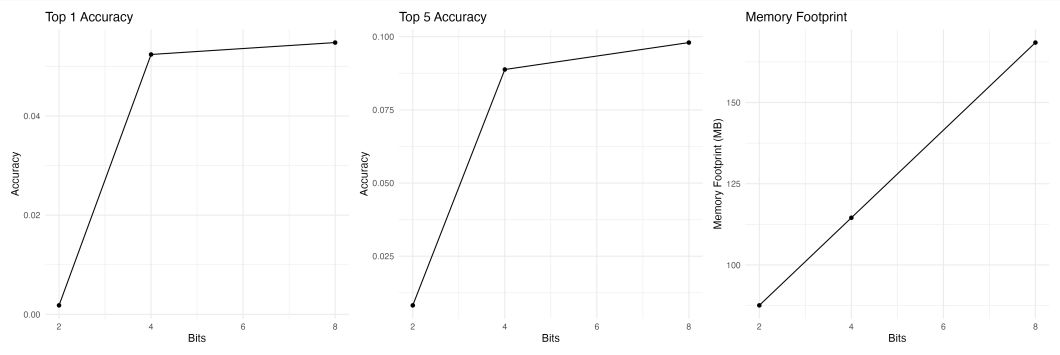


Figure 4: Accuracy vs. Memory Footprint

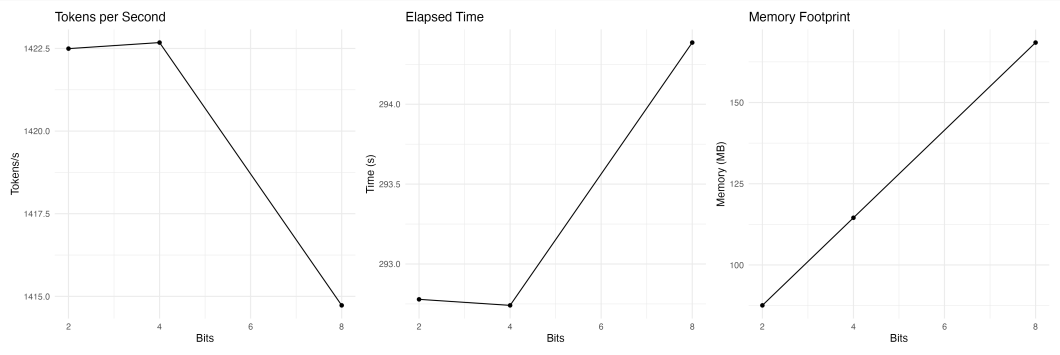


Figure 5: Inference Speed vs. Memory Footprint

Conclusion: experiments demonstrate the viability of using the SmolLM-135M model on resource-constrained devices.

Trade-offs between model size, accuracy, and inference speed make it possible to choose per deployment scenario.

Final Thoughts

“Always Measure One Level Deeper”[1]

We have to get deeper into system behaviors or the underlying factors affecting performance.

This superficiality can lead to incomplete and potentially misleading conclusions about system performance.

[1] Ousterhout, J. (2018). Always measure one level deeper. Communications of the ACM, 61(7), 74-83.

Thank you for your attention!

Source Code: `github.com/sueszli/byte-sized-gains/`

Questions?