Quantisation & Model Compression

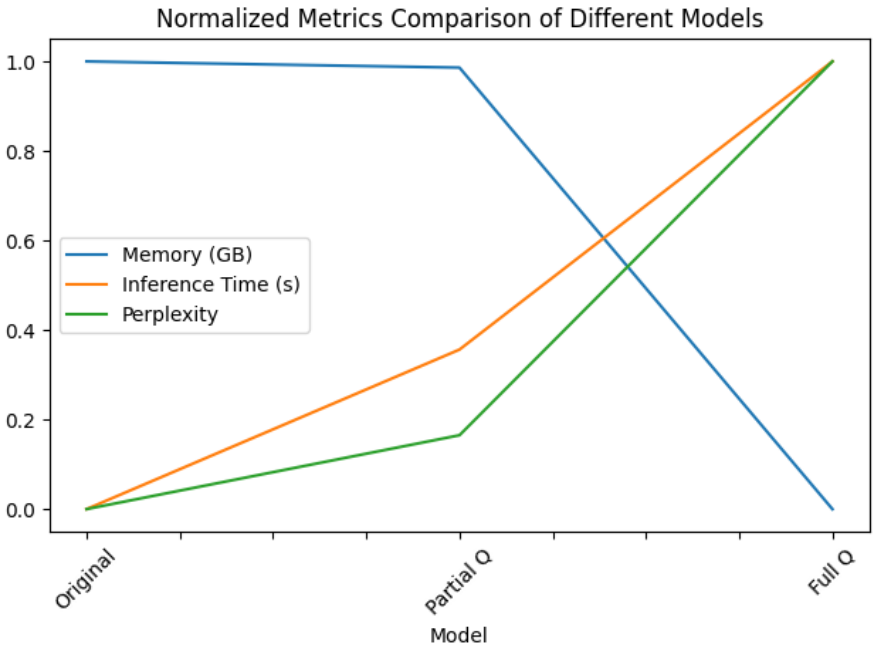
Advanced NLP | Assignment 4

Nanda Rajiv
2021115002

| Part 1: Quantisation from Scratch

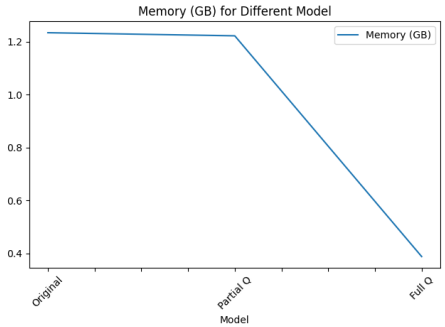
We consider 3 models - the original, partially quantised, and whole model quantised. The partially quantised model has 5 (or a total of 24 layers) quantised.
We observe that there are significant differences between the three models, on the counts of memory, inference time and perplexity.

Analysis and Evaluation



Memory

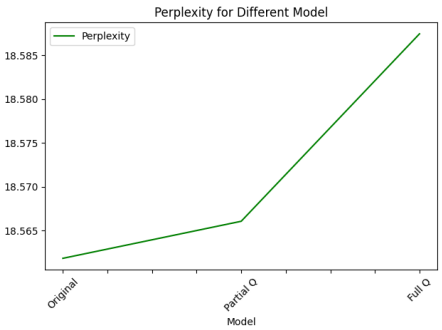
There is a significant decrease in the memory footprint in case of the fully quantised model. In fact, it is about 1/3rd of the the original model.



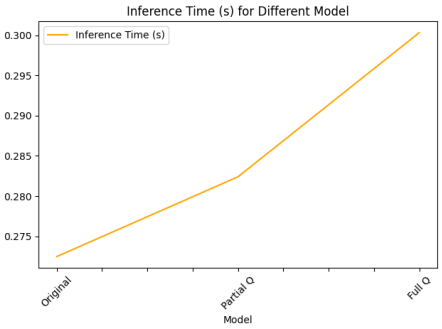
The other two models are fairly comparable in terms of size. This can be attributed to the fact that only 5 layers (of the decoder block) were quantised (of 24 decoder block layers in total, as well as other layers).

Perplexity

We observe that the least perplexity is achieved by the original model. The selectively quantised model has a slightly higher perplexity than this, and the whole quantised model is significantly higher.



Inference Time

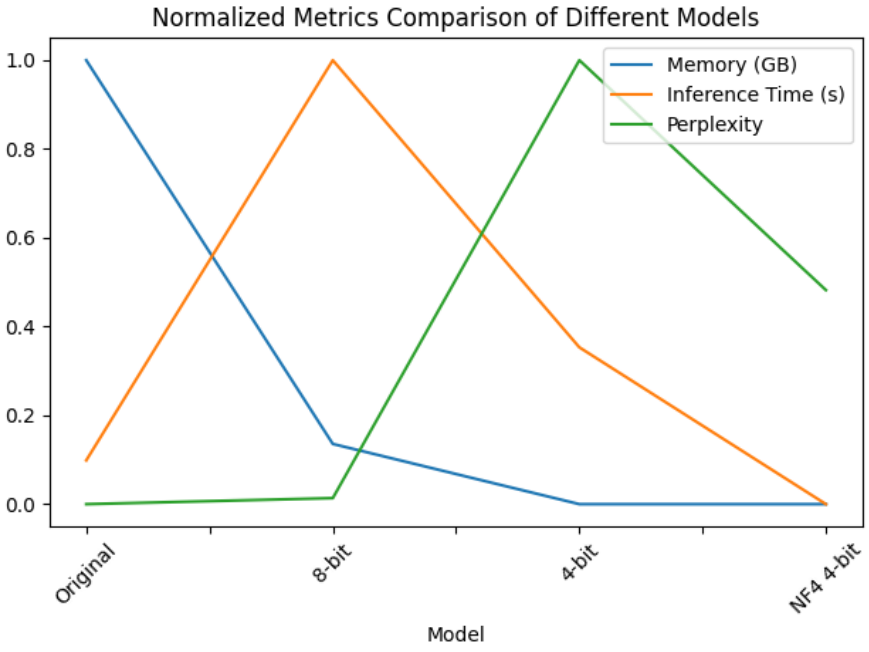


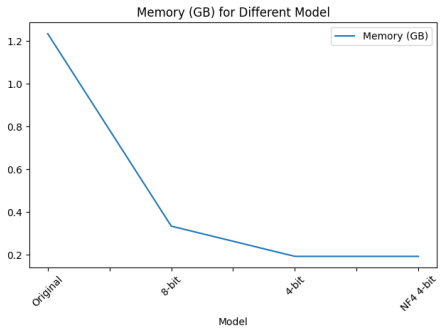
We observe that interestingly the time for the original is lesser than the other two. This seems very counterintuitive.
However, it becomes more clear when considering the fact that my code was run on Ada GPUs. Apparently, GPUs are better optimised for floating point multiplications rather than the int8 or int4 here. This is one potential explanation. There are others too.

Thus, on evaluating using these three aspects, we are likely to opt for a wholly quantised model if we have memory constraints as a primary concern. If not, then one would probably opt for the unquantised on partially quantised model.

| Part 2: Bitsandbytes Integration and NF4 Quantization

Analysis and Evaluation



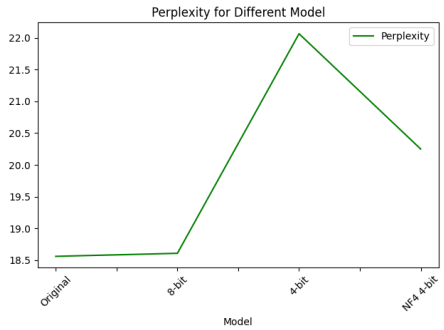


Memory

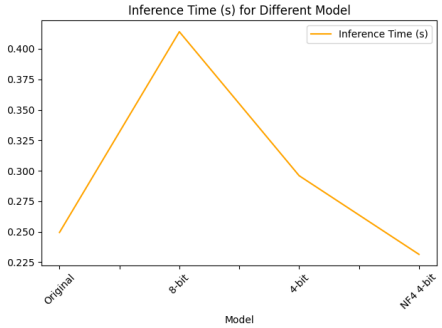
There is a significant decrease in the memory footprint in case of any kind of quantisation. The best memory footprint is achieved through the NF-4 4-bit quantisation, although the linear 4-bit and 8-but are very very close.

Perplexity

We observe that the least perplexity is achieved by the original model. It is, however, very comparable to the 8-but model. The 4-bit model has hte highest perplexities, and the NF-4 model has moderate value.



Inference Time



We observe that interestingly the time for the original is lesser than the 8-bit and 4-bit models. This seems very counterintuitive.
The NF-4 model, however, achieves the fastest inferences, compared to all the other models.

Thus, on evaluating using these three aspects, we are likely to opt for a NF-4 quantised model.

It is worth noting, however, that the differences between this is in fractions of seconds (which were measured average over the whole test dataset). It must be considered too, that in scale, this is not that much of a difference.

| Theoretical Questions

Explain the concept of NF4 quantization and how it differs from linear quantization scales.

Nonlinear Floating-Point 4-bit (NF-4) quantization is a specific type of quantization used that applies a nonlinear mapping to capture the distribution of values, thus improving accuracy at low bit-widths. This is because neural network weights and activations often follow distributions that are not uniform but instead resemble bell curves or have heavy tails. A nonlinear scale provides finer granularity where values are denser (near zero) and coarser granularity in the tails.
Instead of uniformly spacing quantization levels (as in linear quantization), NF4 maps the values logarithmically or using other nonlinear strategies.

Discuss the impact of linear vs. nonlinear quantization on model accuracy and efficiency.

The choice between linear and nonlinear quantization significantly affects both model accuracy and computational efficiency.

The distribution of weights and activations in neural networks plays a critical role in determining the quantization error, which directly impacts accuracy/ Linear quantization divides the range of values into equal intervals, with each interval mapped to a discrete level, while nonlinear quantization adapts intervals based on the data distribution, often using a logarithmic or other nonlinear mapping to allocate more levels where the data is dense. This makes linear strategies a relatively poor fit for non-uniform distributions and a method that gets a higher quantisation loss. However, it is simpler to implement and computationally easier.

When it comes to memory footprint as well, non linear strategies achieve greater memory compression by maintaining acceptable accuracy at lower bit-widths (e.g., 4-bit NF4). Effectively, you can choose lower precision without losing that much accuracy compared to a linear model, thus making it overall more memory efficient.

***