0.1 Evaluation of experimental design

- *Duration of training*
- *volume of data gathered*
- *(im)practicalities - power consumption?*
- *limitations - single optimisation metric*
- *Criticism of methodology*

The size of the pruned networks is not measured.

0.2 Evaluation of results

- *Summary of results per model/dataset*
- *Deep dive into results, detailed visualisations of accuracy & latency tradeoffs (maybe example with poor quality sensitivity analysis vs higher quality layer selection)*
-

We gathered data in 3 phases; a fast pruning phase targeting latency only with no retraining, targeting latency only with retraining, and targeting accuracy only with retraining.

0.2.1 Fast pruning phase

During this phase of the experiment we gathered data to observe how pruning would effect latency, this was useful as an initial proof of concept. This phase of the experiment was very time efficient, we were able to perform 1631 runs with around 18 hours of compute time, each run would usually take between 24-55 seconds. As discussed in section (**TBD**) for this experiment we set the training epochs to 0 and set the target metric to minimize latency.

Interesting observations

- The models that lost all predictive power due to overpruning were not the fastest, even when targeting only latency.

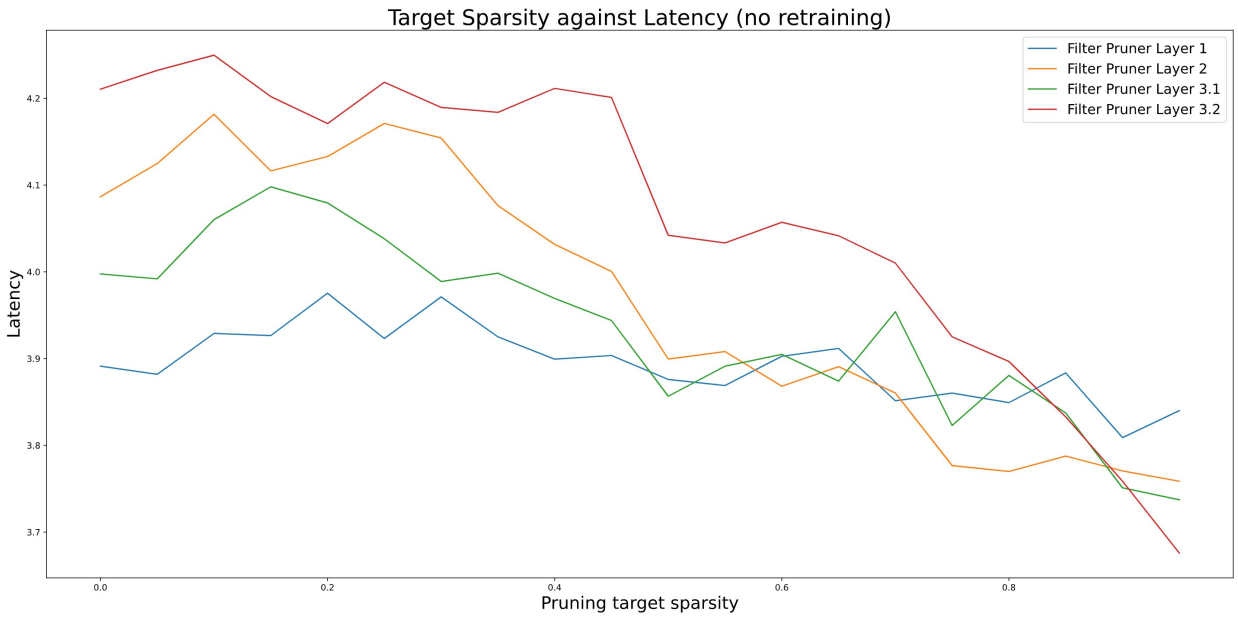

Figure 1: Pruner target sparisity plotted against mean Latency

- The relationship between more pruning and lower latency is not as simple as you get a faster model with fewer tensors
- When targeting accuracy we found models with as low latency when targeting latency directly.
- When targeting latency we found models with as high accuracy as when targeting accuracy directly.