Sparse Upcycling: Training Mixture-of-Experts From Dense Checkpoints

Main Idea: the paper aims to provide an efficient way to train an MoE model from a dense checkpoint (a pre-trained dense transformer) to minimize training costs, that is, provide an MoE training strategy that is cheaper than training from scratch.

- The paper shows that training a MoE from a dense checkpoint outperforms continued dense training.
- Expert-choice routing (with CF of 2) is generally used for the encoder and top-k (with k=2) is used for the decoder.
    - The T5 encoder-decoder model is used as the dense checkpoint.
- Each expert's weights are initialized as the exact MLP of the dense checkpoint, and the router needs to be trained from scratch.
- The layer-norm, attention, embedding and output layers are copied to the new model from the dense checkpoint.

Results:

- When continuing pre-training, the larger the training continues after the checkpoint, the bigger the advantage obtained by the upcycle model vs a dense model.
    - The continued pre-training is referred to as sparse upcycling.
- When sparse upcycling for language, there are two comparisons made:
    - Upcycle vs dense – upcycle performs better, with continued dense pre-training giving inconsistent results.

    - Upcycle vs MoE – upcycle generally performs better for small computational budgets. When enough computational budget is given (>100% of the initial pre-trained dense computational budget), MoE can catch up and perform better than upcycled models.
- Sparse upcycling is also shown to perform better than warm starting ("dense upcycling").

My takeaways:

- It sounds like the approach studied takes T5 (encoder-decoder model) and stretches its feedforward layers horizontally (in other words, transforms them in MoE layers). All other layers remain static – assuming the sparse upcycling is only done on the new MoE layers and routing mechanism, while other layers remain frozen during this process.
- The main takeaway of this paper is that it indicates that with enough training computing budget, it is more efficient to train an MoE model than a dense one, and when not much training computing budget is given, the best-performing approach is to train a sparse upcycled model from a dense checkpoint.