Memorizing Transformer Summary

沙之洲 2020012408

这篇文章的创新点在于，提出了一种在 transformer 中的 memory buffer 的结构，让更深层的 transformer block 能够 attend 到 memory buffer 中的信息。这种结构很大程度上增强了 transformer 在 math problem 上的表现。

这篇文章的优点在于，将 memorizing 的这种结构接到了 transformer 上，同时采用 knn 的方法每次从 buffer 中选取 k 个和当前 attention 层最相关的 memory 进行 attention。这种选取方法没有显著增加 transformer 推理所需要的时间，但是很大程度上提升了 transformer 的推力性能。

这篇文章的局限性在于，在维护 memory buffer 的时候使用的方法过于简单，文章中使用的方法是采取先进先出的原则。这会导致如下的两个问题，第一，有可能前期的信息对于后期的推理是至关重要的，但是先进先出的原则会将这类信息丢掉，一种更好维护 memory buffer 的方法是找到那些最不常用或者最不重要的 memory 丢掉；第二，在向 memory buffer 中加入 attention 向量的时候，没有对向量的重要性进行评估。一些琐碎的，不具有全局性的信息实际上没有必要进入 memory buffer，如何进一步判断哪些向量值得进入 buffer 是一个可以探索的问题。

关于这篇文章的未来方向，我认为它可以被用来构建 AGI 的通用知识库。如果能够开发出 transformer 和 memory buffer 之间的通用接口。那么每一个适配具体任务的，具有 transformer 结构的模型在训练完成后都会有一个 memory buffer。而这个 buffer 相当于模型在任务中学到的知识写成的书籍。如果将这个 memory buffer 接入到另一个没有接触过这个任务的模型上，或许有可能能够让那个模型具备解决这个任务的能力。这就和人类通过书籍学习知识的过程很类似。当人类遇到一个以前没有见过的问题时，通过阅读汲取书籍中记录的知识，人类可以在 zero shot 的情况下具备解决问题的能力。如果能够把这种能力通过 memory buffer 的方式放到 AGI 当中。那么或许更多这对 specific tasks 的能力可以被很快地集成到 LLM 上边，形成在各个任务上全部 sota 的通用智能。