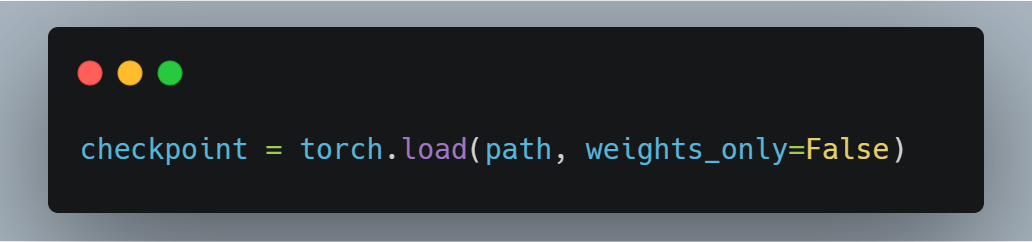
原代码的兼容性问题：

原有代码加载模型时直接使用



对于PyTorch 2.0 及以上的环境，会使用weight\_only参数默认值weight\_only=True，只会加载权重，不会加载完整模型对象。这样会导致后续代码中调用模型checkpoint['model']的操作全部失败（因为只加载了权重，没有模型结构）。

因此需要显式指明weight\_only=False，保证加载检查点时是加载完整的模型对象。



训练优化：

运行原始代码训练过程中，发现GPU大部分时间占用率及功率几乎为0，只有间隔性地出现高峰，且峰值也并不高。经深入研究发现，造成该问题的原因是CPU发送数据的速度跟不上GPU处理数据的速度。于是将DataLoader设置为多线程数据加载，提高了数据加载的速度，减少GPU的等待间隔；并且指定DataLoader将数据加载到固定内存（pinned memory）——GPU 和 CPU 之间数据传输的高效内存区域中，加速数据从 CPU 内存传输到 GPU 显存的过程。通过以上改进，GPU的占用率成功稳定在97%~100%，有效协调了CPU发送数据与GPU处理数据的工作节奏，大大加快了训练速度，数据预处理、预训练、训练阶段的迭代速度都大约提升了一倍。

添加模块：

添加了predict\_with\_trained\_model.ipynb，可以直接调用上述流程训练好的模型，对现有数据或自定义数据进行个性化新闻标题生成任务，并将生成的标题输出到同一目录下的generated\_titles.txt文件中，如果提供了参考标题，则可以自动进行ROUGE评分