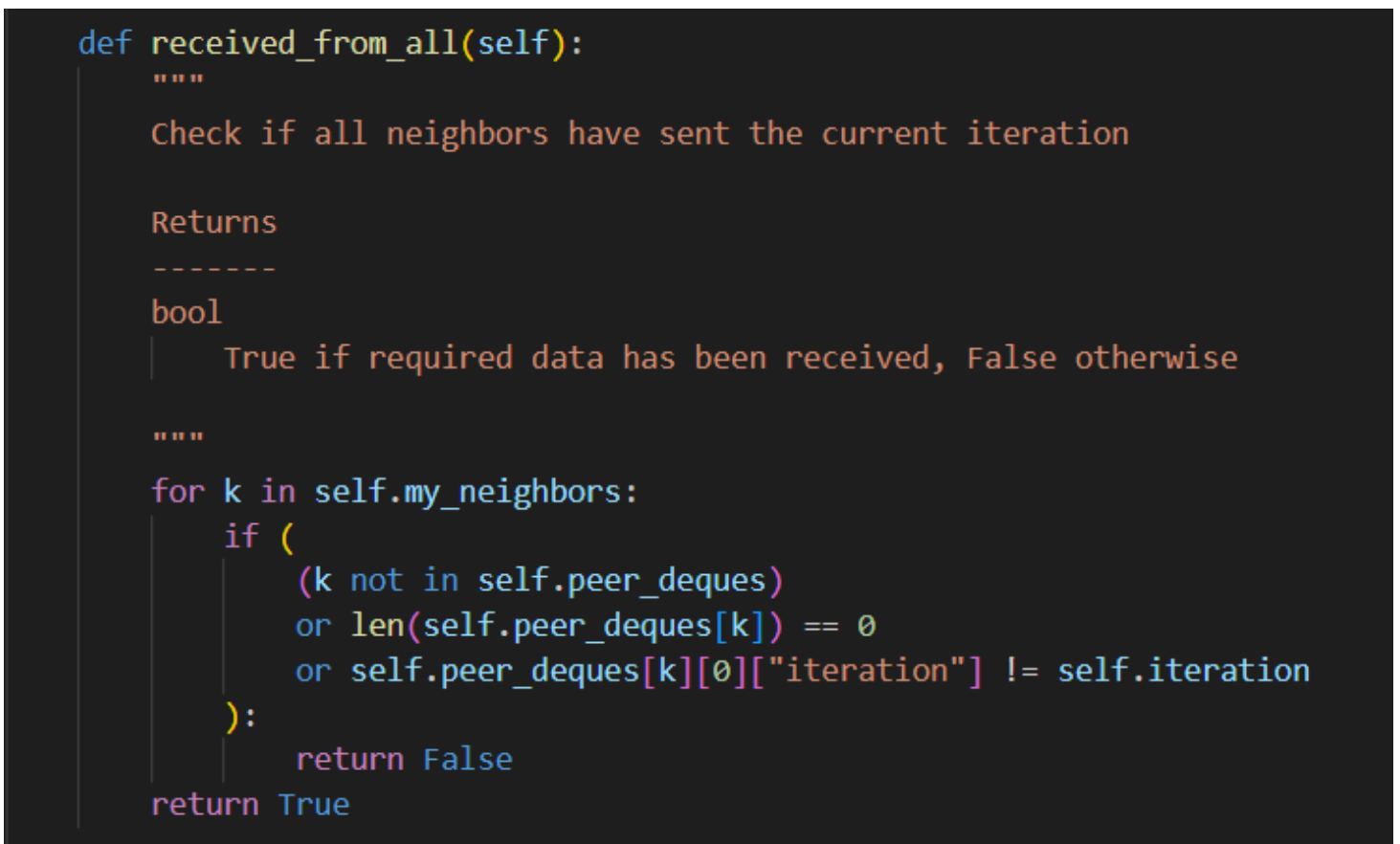


判断是否接受完毕的函数是根据当前轮次，第二轮发送的时候依旧使用 `get_data_to_send` 函数，此时发送的 `data` 中的 `data["iteration"] = self.communication_round` 中的 `communication_round` 已经+1了（`sharing._averaging`函数的副作用，其实也没问题，确实是完成了一轮通信），所以主训练循环的逻辑，`iteration`不应该只+1，而是+2，因为你确实是一个`iteration`传输了两次数据。



不更新iteration的结果就是 `received_from_all` 函数会卡死，因为你发送模型的用户data中的 `iteration` 是 `sharing.communication_round`，但是判断的时候依旧是循环变量 `iteration`，所以用户以为自己迟迟没有收到邻居的数据而陷入死循环等待。