人工智能中的编程第四次作业

本作业旨在通过实现一个简单的深度学习框架来加深对深度学习原理和算法的理解并熟悉一些Python的代码风格。我们将模仿 Needle 框架，分成四个部分完成：运算符的前向运算、梯度计算、拓扑排序以及Tensor类的自动微分以实现一个完整的数据结构。

1. 运算符的前向运算 (`task1_operators.py`)

在这一部分，你需要补充和完善Tensor类，使其能够执行基础的前向运算。Tensor类已经在`task1_operators.py`中被定义，但缺少一些核心的功能。你的任务是实现这些运算符的前向逻辑。

实现步骤：

- 阅读`basic_operator.py`，理解Op类和Value类的基本结构和功能。
- 在`task1_operators.py`中，补充完整Tensor类的运算符，例如乘法、除法、转置、调整形状等。
- 实现后，可以利用`test_task1_forward.py`中的测试函数进行验证。

2. 梯度的反向计算 (`task1_operators.py`)

本部分的目标是实现Tensor类的运算符的梯度计算功能。你需要为每个运算符补充相应的梯度反向计算逻辑。

实现步骤：

- 针对每个运算符，补充其梯度的反向传播公式。例如，对于加法运算，梯度简单地传递到其输入。
- 可以使用`test_task1_backward.py`中的测试函数来验证你的实现。

3. 拓扑排序 (`task2_auto_diff.py`)

在自动微分中，拓扑排序是关键步骤。你可以使用Post-order DFS来进行拓扑排序。

实现步骤：

- 补充实现拓扑排序逻辑，确保正确的计算顺序。
- 完成后，可以使用`test_task2_topo_sort.py`进行测试。

4. Tensor类的自动微分 (`task2_auto_diff.py`, `tensor.py`)

最后一部分是将前面的内容整合，完成一个能够进行自动微分的TensorFull类。你可以使用类似的结构作为Project-Part3的框架。

实现步骤：

- 阅读`task2_autodiff.py`注释，依次完成。
- 可以使用项目中的`test_task2_auto_diff.py`测试代码进行验证。

其他文件

- 基础环境：`basic_operator.py`中定义了Op类和Value类，这是整个框架的基础。在`device.py`中定义了模拟的CPU设备。
- 工具函数： 在`utils.py`中提供了一些可能会用到的工具函数。为了避免交叉引用，可以根据需要将这些函数复制到你的代码中。
- 测试文件：以`test_`开头。在每完成一个部分后，可以使用相应的测试文件进行验证，确保每一步的实现都是正确的。

作业提交

- 本次作业总分10分，会根据完成情况按部分给分。同时请按照作业编排顺序完成，请对于微分图进行显式构建以完成自动微分，给出的测试文件只是参考。我们会根据代码实现进行评分。
- 请于2024.12.01晚23.59分前于course.pku.edu.cn 上提交代码和一个简短的报告说明