**Task2项目第二次报告**

1. **此阶段完成的主要工作**

**结合之前学习的论文内容，分析理解实验代码。**

1. **项目完成内容**

**Analysis experiment code**

1. **实验原理：**

**对于给定的DeepProblog程序，神经网络模型和训练集，首先从外部模型中获取当前参数,然后使用ProbLog 机械计算损失及其梯度,最后使用这些结果来更新神经网络和概率程序中的参数来完成深度学习。**

1. **DeepProbLog 使用的两个模块: ProbLog and PyTorch**

**ProbLog是一种基于Prolog的概率逻辑编程语言；PyTorch是一个开源的Python机器学习库，基于Torch，用于自然语言处理等应用程序。**

**主要代码: gradient\_semiring.py、logic.py、network.py、model.py、optimizer.py、train.py**

1. **代码分析**
2. **gradient\_semiring.py**

**此模块用于计算概率逻辑程序参数的半环梯度（基于Problog）**

**计算函数的传入参数为神经网络模型model和指定的梯度形状的shape，返回结果为概率值和梯度（长度为shape.length的数组）。**

**此模块构造具有规则加法和乘法的概率半线来作为运算符计算SDD上的查询概率。**

1. **logic.py**

**此模块使用Problog处理逻辑程序。**

**extract\_parameters(model)：从外部模型model中提取AD的当前参数。**

**solve(model, sdd, shape)：使用gradient\_semiring模块的方法计算SDD网络的梯度和查询概率。**

1. **network.py**

**此模块构建用于进行深度学习的神经网络。**

**三个神经网络模型：AD\_Network、Fact\_Network、Det\_Network**

**神经网络模型包含有处理规范化、反向传播等功能的函数。**

1. **model.py**

**此模块构建用于训练和测试的概率逻辑模型problog\_model**

**problog\_model由train\_model和test\_model组成，其中train\_model当作外部模型用于相关的参数获取。**

**在构建的model中包含SDD网络的构建，模型梯度和准确度的计算等功能。**

1. **optimizer.py**

**此模块用于概率逻辑模型的优化处理。**

**使用基于标准梯度下降的方法优化了概率逻辑程序和神经网络中的参数，使得概率逻辑模型的学习效果更好。**

1. **train.py**

**此模块用于概率逻辑模型的训练。**

**train函数：**

**对于给定的model和优化器，以及用于训练的查询，计算相应的损失loss。**

**train\_model函数：**

**训练模型，将训练过程的相关数据写入logger中。**

1. **总结**

**下阶段目标：运行三个example，分析实验结果，并改写MNIST减法。**