
一、深度学习科研代码管理和实验管理面临的问题

深度学习模型结构变化程度大、参数多，如何高效地进行代码和实验管理？

1. **环境管理**：如何保证代码运行环境可复现；
2. **模块管理**：如何划分模型的核心和任务，以及一些辅助的目录；
3. **分支管理**：如何划分 git 分支；
4. **大文件管理**：如何管理数据集、模型文件等大型二进制文件；
5. **参数管理**：如何管理训练参数；
6. **结果管理**：如何管理日志、输出数据和实验结果；
7. **进度管理**：如何可视化训练进度、训练用时；
8. **随机数管理**：如何处理实验的 random seed；
9. **错误处理**：如何处理 Python 中的错误，避免实验中断；
10. **调试模式**：如何实现小数据集的调试模式，避免执行用时过长；
11. **图表管理**：如何管理论文中可能用到的文档图表；
12. **文档管理**：如何编写和管理文档，包括笔记、论文和 slides。

二、环境管理

使用 `requirements.txt` 和 `miniconda` 来管理项目的环境。

conda 新建环境：

```
conda create -n envName python=3.9
```

conda 查看所有环境：

```
conda env list
```

conda 删除环境：

```
conda env remove -n envName
```

生成 `requirements.txt` 文件：

```
pip freeze > requirements.txt
```

安装 `requirements.txt` 依赖：

```
pip install -r requirements.txt
```

使用 conda 安装 requirements.txt 依赖：

```
conda install --yes --file requirements.txt
```

使用 conda 安装 requirements.txt 依赖（失败时用 pip 重试）：

```
while read requirement; do conda install --yes $requirement || pip install $requirement; done < requirements.txt
```

三、 模块管理

良好的目录划分和模块划分是可维护性的关键。一般可以分为这几大块：

数据	稳定代码	易变代码
<div>原始数据</div> <div>预处理数据</div>	<div>I/O 模块</div> <div>预处理模块</div>	<div>实验参数配置</div>
<div>模型文件</div> <div>日志数据</div>	<div>可视化模块</div>	<div>训练/测试脚本</div>
<div>实验数据</div> <div>实验结果</div>	<div>数据加载模块</div> <div>模型主体</div>	
<div>文档</div>	<div>损失函数</div> <div>后处理模块</div>	
	<div>训练函数模块</div>	

四、 分支管理

我们应该使用 git 分支管理管理好我们的主分支和试验分支。