

整定计算微调大模型

一、哪些环节需要微调？哪些绝对不能用？

业务环节	是否需要微调大模型	原因与技术替代方案
设备参数 OCR /文档提取	非常需要(小模型或多模态大模型)	电网设备说明书、测试报告格式繁杂, 表格错位严重。通过微调(SFT)通用多模态模型(如 Qwen2-VL / MiniCPM-V), 能极其精准地把 PDF/扫描件里的“短路阻抗”、“变比”等字段提取成结构化 JSON。
定值单模板推荐	不需要	使用 规则引擎 + 向量数据库(RAG) 即可, 微调模型不仅成本高, 而且容易产生不可控的推荐错误。
定值整定计算与配合	绝对不能	大模型天生缺乏严格的数学逻辑和符号推导能力, 微调也无法解决“幻觉”和计算失真问题。必须用 Python/C++ 编写确定性的电力规则公式与算法。
定值书/计算报告生成	可选/推荐微调	供电局对于定值书的文字表达、异常说明有特定的“体制内/电网术语”风格。微调轻量级开源大模型(如 Qwen2.5-7B/14B), 能让自动生成的文字描述非常专业、符合电网规范。

二、如何准备微调数据？

数据准备是微调中最耗时、也最关键的一步(通常占微调工作量的70%以上)。针对电网这个场景

，主要准备两类数据：

1. 结构化抽取类数据(用于设备参数提取)

- 原始数据收集：收集 300~500 份真实的配电设备说明书、巡检报告、铭牌照片、PDF/Word 文档。
- 标注格式设计：整理出需要抽取的字段清单(如：设备型号、额定电压、CT变比、零序阻抗等)。
- 构建训练对(JSON/Prompt格式)：
 - **Input (输入)**: 经过 OCR 识别后的文本片段或直接传入文档图片。
 - **Output (期望输出)**: 标准的 JSON 数据。

JSON

```
"instruction" "请从以下配电设备文档片段中, 提取设备名称、型号、额定电压和变比。"

"output" "{\n  \"device_name\": \"开关柜\", \n  \"model\": \"KYN28A-12\", \n  \"rated_voltage\": \"10kV\", \n  \"ct_ratio\": \"600/5\" \n}"
```

2. 文本生成类数据(用于定值书/报告生成)

- 原始数据收集：收集 100~200 份供电局历史优秀的定值书、整定计算说明书范本。
- 构建训练对：
 - **Input (输入)**: 计算结果数据 + 设备参数(JSON/表格)。
 - **Output (期望输出)**: 符合电网公文/技术规范的段落文本。

JSON

```
"instruction" "根据提供的定值计算结果和配合关系, 生成过流一段保护的整定说明文本。"

"output" "本线路10kV出线开关过流速断保护按躲过线路末端最大短路电流整定, 定值拟定为12.5A, 动作时限取0.3s, 以实现与下级变压器高压侧保护的时限级差配合....."
```

三、具体的微调流程与落地步骤

在私有化部署或本地开发环境中, 微调流程通常分为以下 5 个步骤:

[1. 数据清洗与标注] → [2. 选择基座模型] → [3. 采用LoRA轻量化微调] → [4. 模型评估与幻觉测试] → [5. 模型部署与API封装]

1. 选择基座模型(Base Model):

- 文本生成/抽取类推荐使用: **Qwen2.5-7B-Instruct** 或 **Qwen2.5-14B-Instruct**(中文能力强, 适合电网术语)。
- 多模态图文抽取推荐使用: **Qwen2-VL-7B**。

2. 选择微调框架与方法：

- 采用 **LoRA (Low-Rank Adaptation)** 或 **QLoRA** 技术, 这样不需要训练整个模型的几百亿参数, 只需训练很小一部分附加参数。
- 工具推荐使用 **LLaMA-Factory** 或 **Unsloth**, 这些开源工具提供了图形界面或简单的命令行, 无需从零写深度学习训练代码。

3. 训练算力需求：

- 使用 7B/14B 模型 + LoRA 微调, 单张消费级显卡(如 **RTX 4090 24GB**)或单张 **A10/A800** 即可在 2~4 小时内完成几百条数据的微调, 硬件门槛和成本低。

4. 模型评测与边界校验：

- 微调完成后, 必须用未参与训练的 50 份新文档进行测试(计算 Precision/Recall 召回率)。
- 针对抽取结果, 必须在代码层加入 **Schema** 校验(格式检查)与范围合法性校验(例如变比不能为负数), 如果校验失败则触发人工二次确认。

四、实施建议

1. 不要为了微调而微调：优先尝试用 **Prompt 工程 + RAG**(检索增强生成)解决。只有当单纯靠写 Prompt(提示词)无法保证设备参数提取格式 100% 稳定时, 再启动微调。
2. 微调只做“翻译”和“格式化”：让微调大模型扮演“把乱七八糟的文档翻译成标准JSON”或者“把冷冰冰的数字翻译成电网报告”的角色, 核心计算逻辑一律交给 Python 确定性代码。