**智慧政法讲义**

# 1法律行业业务背景

随着社会的发展和法律制度的完善，人们对法律咨询的需求日益增长。法律领域的知识十分复杂，包括各个法律领域的法规、判例、先例等。法律咨询需要考虑到各种法律细节和情境，并结合具体案件的事实进行分析和判断。因此无论是个人还是企业，他们在面临法律问题时需要专业的意见和指导。然而，律师的数量有限且收费昂贵，无法满足所有人的需求。因此，开发一种能够提供准确、及时法律咨询的大模型是非常必要的。

法律大模型可以通过在线平台或聊天机器人等形式，为广大民众提供普惠的法律咨询服务。无论是面对个人法律问题还是企业法律难题，用户可以随时随地获取准确、快速的法律咨询，提高法律意识和保护自身权益。法律大模型也可以承担部分法律咨询的工作，减轻法律专业人员的负担。法律大模型还可以处理常见问题和基本法律知识，为法律从业者提供支持和参考，使他们能够更好地应对复杂的法律案件和争议。

法律大模型还具有以下应用场景：

* 答辩状总结提取：对冗长的答辩状做精简化总结
* 争议焦点分析：根据起诉状和答辩状等文件内容，让大模型根据多份文件自动分析争议焦点
* 证据分析：根据已有证据和案件详情判断嫌疑人是否具有刑事行为能力

法律大模型的出现为给行业所能带来的“惊喜”也重点体现在以下几个层面：

首先是可以提供全流程智能辅助办案应用，构建智能审查、量刑预测、文书生成、自动编目、笔录生成等业务能力，通过智能技术辅助办案人员的常规工作。

二是提供全方位司法监督管理应用，基于大模型建设视频自动巡查、案件裁判偏离预警、案件智能核查等监督助手应用，及时发现和解决问题，加强司法工作流程的规范化。

三是提供司法大数据深度挖掘应用，快速在海量的法律文本中搜索获取相关案例、法规和法律文献等信息，探索司法规律和趋势，为司法改革和法律制定提供数据支持。

除了针对从业人员的智慧化辅助外，大模型的通用能力对于社会大众也有广阔的应用前景，对于获取更高效的司法服务是变革性的提升。

总之，通过将自然语言处理技术应用于法律领域，可以改善法律服务的效率和质量，提高法律体系的智能化水平。同时，大模型的应用还可以为法律数据分析、司法决策等方面提供有力支持。

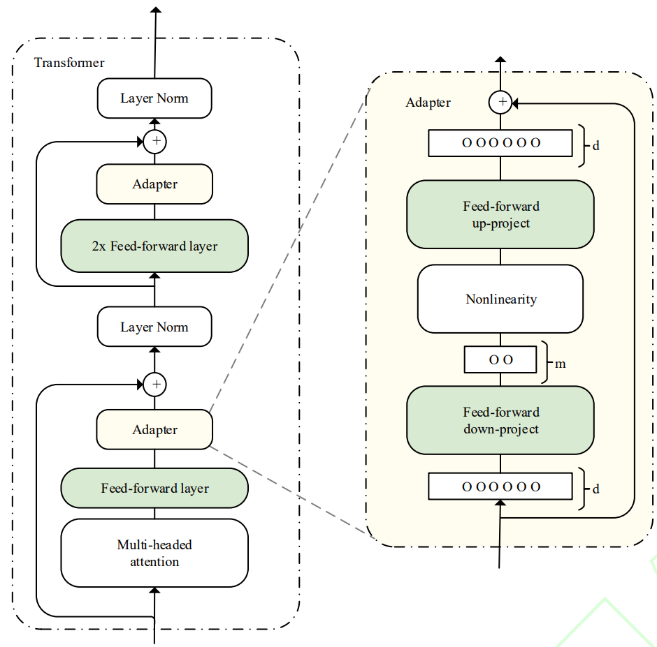
# 2大模型微调技术

大模型拥有庞大的参数量，全量微调需要消耗大量的算力资源，因此需要采用一些低资源微调技术来减少算力资源的消耗。目前，常见的微调方法有Adapter微调、Prefix tuning和Lora微调。

## 2.1 Adapter

Adapter Tuning 在预训练模型每层中插入用于下游任务的参数(针对每个下游任务， 仅增加 3.6% 的参数)， 在微调时将模型主体冻结，仅训练特定于任务的参数，从而减少了训练时的算力开销。

Adapter Tuning 设计了 Adapter 结构， 并将其嵌入 Transformer 的结构里面， 针对每一个 Transformer 层， 增加 了两个 Adapter 结构 (分别是多头注意力的投影之后和第二个 feed-forward 层之后)，在训练时， 固定住原来预训 练模型的参数不变， 只对新增的 Adapter 结构和 Layer Norm 层进行微调， 从而保证了训练的高效性。每当出现新 的下游任务，通过添加 Adapter 模块来产生一个易于扩展的下游模型，从而避免全量微调与灾难性遗忘的问题。



上图是 adapter 的具体结构， 每个 Adapter 模块主要由两个前馈(Feedforward) 子层组成， 第一个前馈子层 (down-project) 将Transformer块的输出作为输入，将原始输入维度d (高维特征) 投影到 m (低维特征)，通过控制m的大小来限制Adapter模块的参数量，通常情况下，m«d。然后，中间通过一个非线形层。在输出阶段，通过第二个前馈子层(up-project) 还原输入维度，将m (低维特征) 重新映射回d (原来的高维特征)，作为Adapter模块的输出。同时，通过一个skip connection来将Adapter的输入重新加到最终的输出中去，这样可以保证，即便Adapter一开始的参数初始化接近0，Adapter也由于skip connection的设置而接近于一个恒等映射，从而确保训练的有效性。

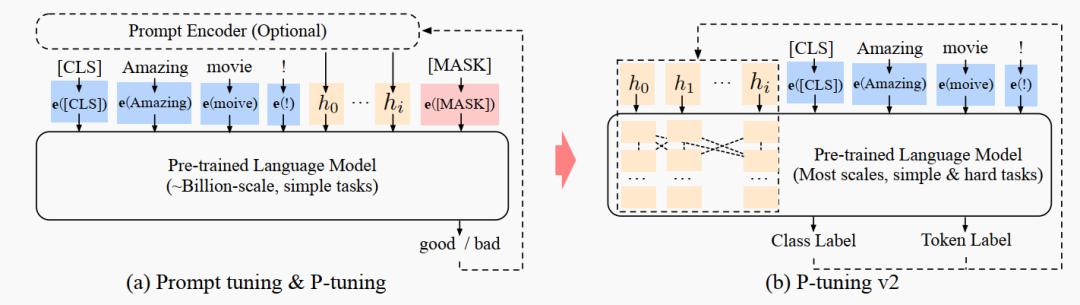
至于adapter引进的模型参数，假设adapter的输入的特征维度是d，而中间的特征维度是m，那么新增的模 型参数有：down-project 的参数 d\*m+m，up\_project的参数 m\*d+d，总共 2md+m+d， 由于m远小于d，所以真实情况下，一般新增的模型参数都只占语言模型全部参数量的 0.5%~8%。同时要注意到，针对下游任务训练需要更新的参数除了adapter引入的模型参数外，还有adapter 层后面紧随着的 layer normalization 层参数需要更新，每个 layer normalization 层只有均值跟方差需要更新，所以需要更新的参数是 2d。

通过实验发现，只训练少量参数的Adapter方法的效果可以媲美全量微调，这也验证了 Adapter是一种高效的参数训练方法，可以快速将语言模型的能力迁移到下游任务中去。Adapter通过引入0.5% ~ 5%的模型参数可以达到不落后全量微调模型1%的性能。

## 2.2 P-tuning

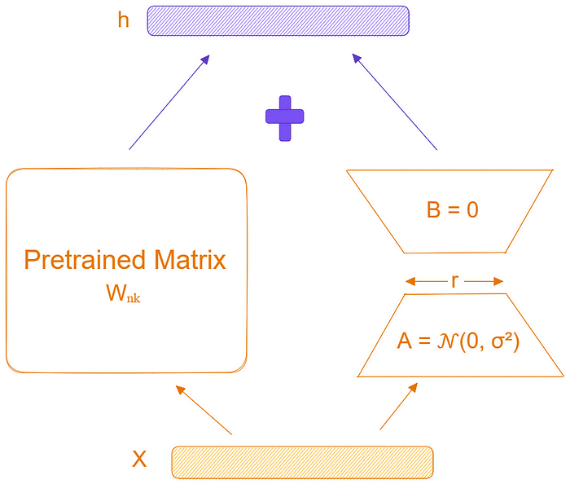
P-tuning是一种soft prompt方法，大模型对prompt比较敏感，通过人工撰写难以找到合适的自然语言的prompt。相比Prefix Tuning，P-Tuning加入的可微的virtual token，但仅限于输入层，没有在每一层都加；另外，virtual token的位置也不一定是前缀，插入的位置是可选的。这里的出发点实际是把传统人工设计模版中的真实token替换成可微的virtual token。因此，P-tuning的思想是让模型自己寻找prompt，在微调时只需要更新这部分参数即可。

P-Tuning v2是在P-Tuning的改进，该方法在每一层都加入了Prompts tokens作为输入，而不是仅仅加在输入层，这带来两个方面的好处：更多可学习的参数（从P-tuning和Prompt Tuning的0.01%增加到0.1%-3%），同时也足够参数高效。加入到更深层结构中的Prompt能给模型预测带来更直接的影响。



## 2.3 LoRa

LoRA技术使用秩分解来约束更新矩阵 ΔW 的秩，它将 ΔWₙₖ 表示为 2 个低秩矩阵 Bₙᵣ 和 Aᵣₖ 的乘积，r<<min(n,k)。A使用随机高斯初始化，B初始为0。这样在训练时训练的参数量从（nk）减少到r（n+k），大大减少了模型微调所需要的时间与空间。另外，更新矩阵的向量可以单独存储，在推理时只需和基座模型合并即可，无需消耗额外的推理时间，同时面对不同的任务只需切换不同lora权重即可。



# 3 Gradio框架

Gradio基于svelte，它的封装组件的功能相对比较完整，可以让开发者专注于编写基于 python 代码的业务处理逻辑，无需关注更多 web 前端页面的实现细节。Gradio封装程度高，可以快速构建一个大模型交互页面，可以快速创建共享连接进行分享，也可以在jupyter notebook中展示。但是组件的扩展性较差，如果想自定义组件就得修改源码，同时可能无法满足一些复杂的场景下的应用。