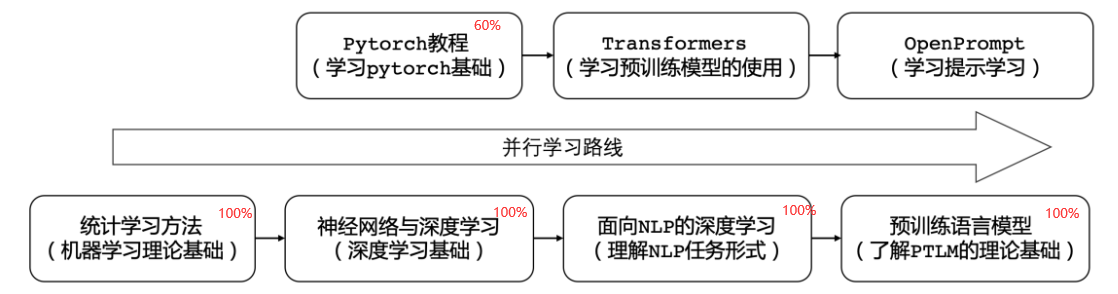
**学习进度**



**四、自然语言处理中的预训练模型**

1. 自然语言处理的基础是语言表示。早期方法是知识库规则的形式，随着深度学习在自然语言处理中的应用，目前比较流行的方式是分布式表示，即把语言的语义分散在一组不同维度的向量中，该过程也称为嵌入（Embedding）。表示学习在自然语言处理任务中起到非常重要的作用。
2. 和人类学习语言类似，我们希望机器学会一种通用的语言表示。预训练模型一般比随机初始化更好，预训练可以视为一种正则化方法，下游任务的参数会受到预训练模型参数的约束。第一代预训练模型围绕非上下文向量展开，比如word2vector（词的向量化），glove。而第二代预训练模型在此基础上关注了上下文向量，如BERT。预训练任务可以分为监督学习、自监督学习、无监督学习。监督学习需要有大量标注数据的监督任务，常用于机器翻译和机器阅读理解任务，比如Context Vectors (CoVe)。无监督学习能够用于预训练的两类任务是密度估计（（统计）语言模型）和重构（自编码器）。
3. **自监督学习**可以看作一种监督学习和无监督学习的结合，使用监督式的学习方式，训练数据由无标注数据自动构建。自监督学习的核心是不依赖标注数据，伪监督任务，学习目标并不是在自监督任务上学习地好，而是学习数据中的可泛化知识，从而应用于下游任务。
4. 我们希望将预训练模型学习到的知识迁移到下游任务。目前主流的范式是**预训练+精调**，即首先用大规模标注数据预训练Transformer模型，然后用少量标注数据在下游任务上精调。在选择预训练模型时，首先考虑模型架构，我们要选择一个和下游任务相似的预训练模型，比如BERT作为一个Encoder不适合生成任务，T5和BART更适合做生成任务。如果下游任务是跨语言的，那就选择跨语言预训练模型效果会更好。其次是考虑预训练任务，不同的预训练任务有自己的偏向，对不同的任务有不同的效果。最后，我们希望预训练阶段和下游任务的数据类型要尽可能类似。

# 五、Pytorch 教程

本周学习了PyTorch介绍部分，对重点知识做了笔记。计划下周完成Text部分。