Drylab报告

1. 数学定义

设 $x = (x_1, x_2, \ldots, x_n)$ 表示一个 DNA 序列，其中每个 $x_i \in \{A, C, G, T\}$。我们希望设计一个映射函数

$$f : \mathcal{X} \to [0, 1]$$

使得 $f(x)$ 能够输出一个概率值，表示该序列为启动子的可能性，其中 1 表示启动子，0 表示非启动子。

目标是在给定训练样本 $\{(x^{(i)}, y^{(i)})\}_{i=1}^{N}$ 的情况下，最小化损失函数

$$L(\theta) = -\frac{1}{N} \sum_{i=1}^{N} \left[y^{(i)} \log\Big(f(x^{(i)}; \theta)\Big) + (1 - y^{(i)}) \log\Big(1 - f(x^{(i)}; \theta)\Big) \right]$$

其中 θ 为模型参数。通过求解

$$\min_{\theta} L(\theta)$$

我们可以获得一个能够自动判定 DNA 序列是否含有启动子区域的模型。

---

2. 数据与方法

数据来源与预处理

- **数据来源**:
  我们使用了 HuggingFace 上公开的 **dna_core_promoter** 数据集，该数据集包含标注了 DNA 序列及其对应是否为启动子的标签（0 或 1）。
- **预处理步骤**:
  1. **字符映射**：将 DNA 序列中的字符 A、C、G、T 分别映射为整数（例如 A→1, C→2, G→3, T→4），其中 0 用作填充符号。
  2. **序列填充**：对长度不足的序列进行零填充，使所有序列达到统一长度。

3. **数据划分**：将数据划分为训练集和测试集（80%/20% 分割）。

方法与模型架构

我们采用深度学习方法构建了一个基于 LSTM 和注意力机制的模型，其主要结构如下：

```
        输入 DNA 序列
              |
       [字符索引映射]
              |
       [嵌入层 (Embedding)]
              |
      [双向 LSTM 层]
              |
      [注意力机制层]
              |
      [全连接层 (FC)]
              |
        输出：启动子概率
```

- **嵌入层**：将输入的字符索引映射到一个连续的向量空间，从而捕捉核苷酸的语义信息。
- **双向 LSTM 层**：提取序列中的时序依赖信息，双向结构可以同时捕捉正向和反向的上下文信息。
- **注意力机制**：在 LSTM 的所有时间步输出上计算注意力权重，通过加权求和得到一个聚合特征表示，自动聚焦于对分类最有贡献的部分。
- **全连接层**：将聚合后的特征输入全连接层，并通过 Sigmoid 函数输出一个在 [0,1] 区间的概率值，作为分类依据。

---

3. 分析和解释结果

- **模型性能与准确率**
  虽然模型在测试集上的准确率大约达到 80%~85%。
- **注意力机制的作用**
  通过引入注意力层，模型能够自动为序列中的某些区域分配较高的权重。这意味着模型在判定是否存在启动子时，会重点关注某些特定位置的核苷酸组合。进一步的可视化分析（例如注意力权重图）显示，这些区域往往与生物学上已知的重要调控区域相吻合。

- **数据与模型的局限性**
  - 数据集样本数量有限，可能限制了模型学习更复杂特征的能力。
  - 某些序列中的低复杂度区域或噪声数据可能会导致误判，提示在未来可以考虑数据增强或更复杂的预处理方法。
  - 目前的模型结构虽然能够捕捉较为丰富的时序信息，但对超参数（如 LSTM 层数、隐藏单元数、注意力机制的具体实现）还有进一步优化的空间。