**2. RNN、LSTM 和 GRU 模型解析**

在深度学习中，RNN（循环神经网络）、LSTM（长短时记忆网络）和 GRU（门控循环单元）是用于处理时序数据的重要模型。它们主要用于自然语言处理、时间序列分析等任务。

**(1) RNN（Recurrent Neural Network）**

* RNN 适用于处理序列数据，能够保留之前的状态信息，并将其传递到后续步骤。
* 其核心是一个隐藏状态 hth\_t ，通过一个循环结构不断更新：

其中，和是网络权重，是当前输入，是前一时刻的隐藏状态。

* **缺点**：
  + **梯度消失和梯度爆炸问题**：随着序列增长，远处的信息难以影响当前状态，使得 RNN 在长序列学习时表现较差。

**(2) LSTM（Long Short-Term Memory）**

* LSTM 是为了解决 RNN 的长距离依赖问题而设计的，能够较好地缓解梯度消失问题。
* 主要由 **输入门、遗忘门、输出门** 组成，每个时间步 tt 的更新如下：
  + **遗忘门**：决定保留多少过去的信息。
  + **输入门**：决定当前信息如何更新记忆。
  + **输出门**：决定当前隐藏状态的输出。

其核心计算公式：

其中：

* + 是遗忘门
  + 是输入门
  + 是输出门
  + 是记忆单元状态
* **优点**：
  + 通过门控机制有效保留长期依赖信息。
  + 适用于处理长时间序列的数据。

**(3) GRU（Gated Recurrent Unit）**

* GRU 是 LSTM 的一个变体，结构更简单，计算效率更高。
* 仅由两个门 **更新门和重置门** 组成：

其中：

* + 控制隐藏状态的更新量（类似 LSTM 的输入门和遗忘门的结合）。
  + 控制前一个状态的信息是否被遗忘。
* **优点**：
  + 结构比 LSTM 更简单，计算速度更快。
  + 在许多任务中，GRU 具有与 LSTM 相似的性能，但参数较少。

**3. 诗歌生成的过程**

**(1) 数据准备**

* 代码中的 gen\_data\_batch 用于生成数据样本，主要是随机生成两个整数，并计算它们的和。
* convertNum2Digits 用于将整数转换为按位拆分的列表，便于后续处理。
* prepare\_batch 负责将数据转换为合适的格式，包括：
  + 将整数拆分成数位。
  + 反转数位顺序，使低位优先。
  + 用 0 填充不足长度的数列，确保固定长度。

**(2) 构建 RNN 模型**

* myRNNModel 是一个 RNN 模型：
  + **嵌入层 (Embedding)**：将输入数值映射为低维向量（这里是 10 维输入，映射到 32 维）。
  + **RNN 层 (SimpleRNNCell)**：包含一个 64 维的隐藏层。
  + **Dense 层**：全连接层，将 RNN 输出映射回 10 维（数字 0-9）。

**(3) 训练过程**

* train\_one\_step 计算损失，并通过 GradientTape 计算梯度并更新模型参数。
* train 负责完整的训练过程，每 50 步打印一次损失，评估模型性能。

**(4) 评估**

* evaluate 通过模型预测结果，并转换回整数，计算准确率。
* 评估时，生成新数据并通过训练好的 RNN 模型预测加法结果，计算正确率。

**总结**

* 代码的主要任务是 **使用 RNN 预测整数加法**，类似于学习数字序列的模式。
* 采用 **简单 RNN** 结构，但可以扩展为 **LSTM 或 GRU**，提高长期依赖的学习能力。
* 训练时，模型学习如何将数位逐步输入，并预测正确的结果。
* 这种方式可以用于诗歌生成任务，例如：
  + 通过 LSTM/GRU 处理文本序列，学习诗歌的模式。
  + 通过 Embedding 层将字符转换为向量表示。
  + 通过 RNN 逐步预测下一个字或词，实现诗歌自动生成。

4. 生成诗歌



5. 总结

本实验基于 RNN 训练了一个诗歌生成模型，完成了数据预处理、模型构建、训练优化，并成功生成以“日、红、山、夜、湖、海、月”等词汇开头的诗句。训练过程中采用 Adam 优化器和交叉熵损失函数，经过 10 轮训练，loss 逐步降低，模型学会了基本的诗歌语言模式。实验结果表明，模型能够生成符合古诗风格的句子，但仍存在语义不连贯、部分重复等问题。改进方向包括使用 LSTM/GRU 替代 SimpleRNN、优化采样策略（如 Top-k 采样）、扩大训练数据集，甚至尝试 Transformer 结构，以提升诗歌生成质量和可读性。