# NKU 深度学习（高阶课）实验报告


实验名称：注意力机制

学　　院：网络空间安全学院

姓　　名：田晋宇

专　　业：物联网工程

二〇二五年六月

# 目录

# 1 实验要求

- 掌握注意力机制的基本原理，理解其在神经网络中如何提升长序列建模能力。
- 学会使用 PyTorch 框架构建一个基于注意力机制的 Seq2Seq 模型。
- 实现法语到英语的自动翻译功能，验证注意力机制在机器翻译任务中的效果。

# 2 基于 RNN 解码器的 Seq2Seq 模型

本节介绍基于门控循环单元 GRU 的序列到序列 Seq2Seq 模型，主要由编码器（EncoderRNN）和解码器（DecoderRNN）两个部分组成。该模型广泛应用于机器翻译、对话生成等自然语言处理任务，其核心思想是将变长的输入序列编码为定长的上下文向量，再逐步生成目标序列。

## 2.1 编码器结构

编码器模块 `EncoderRNN` 主要包含以下组件：

- **嵌入层（Embedding Layer）**：`nn.Embedding(input_size, hidden_size)`，将输入的词汇索引序列映射为稠密的词向量。
- **Dropout 层**：用于防止过拟合。
- **GRU 层**：`nn.GRU(hidden_size, hidden_size, batch_first=True)`，用于对嵌入后的序列进行时序建模。

在训练阶段，输入张量形状为 $(N, L)$，其中 $N$ 为 batch size，$L$ 为输入序列长度。嵌入后维度变为 $(N, L, H)$，其中 $H$ 为 hidden size。经 GRU 处理后，输出张量为 $(N, L, H)$，同时返回最终时间步的隐藏状态 $(1, N, H)$，作为解码器的初始隐状态。

## 2.2 解码器结构

解码器模块 `DecoderRNN` 包含以下组成部分：

- **嵌入层**：将当前输入词映射为向量形式。
- **GRU 层**：接收当前嵌入表示与前一时刻隐藏状态，更新当前隐藏状态。
- **输出线性层**：`nn.Linear(hidden_size, output_size)`，将 GRU 输出投影至词汇表大小的 logits 空间。

解码器以时间步 $t$ 为单位进行迭代生成。每一轮，输入为 $(N,1)$ 的 token 索引，经嵌入层变为 $(N,1,H)$，传入 GRU 并结合上一隐藏状态 $(1,N,H)$，输出新的隐藏状态及 GRU 输出 $(N,1,H)$，再通过线性层变换为 $(N,1,V)$，其中 $V$ 为词汇表大小。

整个解码过程运行 $T$ 个时间步，$T$ 可为目标序列最大长度或解码终止条件。所有时间步输出沿时间维拼接，形成最终输出张量 $(N,T,V)$，并通过 `log_softmax` 进行归一化，用于训练的负对数似然损失计算。

### 2.3 训练阶段的输入输出维度变化

训练过程中，编码器与解码器的数据维度变化如下所示：

| 模块 | 输入维度 | 输出维度 |
|---|---|---|
| Embedding（Encoder） | $(N,L)$ | $(N,L,H)$ |
| GRU（Encoder） | $(N,L,H)$ | 输出: $(N,L,H)$; 隐状态: $(1,N,H)$ |
| Decoder Input | $(N,1)$ | 作为每轮输入（初始为 `<SOS>`） |
| Embedding（Decoder） | $(N,1)$ | $(N,1,H)$ |
| GRU（Decoder） | $(N,1,H)+(1,N,H)$ | $(N,1,H)+(1,N,H)$ |
| Linear | $(N,1,H)$ | $(N,1,V)$ |
| 拼接所有时间步 | $T$ 个 $(N,1,V)$ | $(N,T,V)$ |

表 1: 训练阶段输入输出维度变化

基于 RNN 的 Seq2Seq 模型结构清晰，能够实现输入序列到输出序列的映射。编码器通过 GRU 捕捉上下文语义信息，解码器则逐步生成输出，支持 Teacher Forcing 等训练技巧。其模块化设计便于扩展，如引入注意力机制、变换为双向编码器、多层 GRU 结构或替换为 Transformer 架构等。通过明确的输入输出维度控制，该结构可高效支持小规模实验与复杂任务的建模，是神经序列建模的基础方法之一。

### 2.4 实验结果及分析

为了解 Seq2Seq 模型在未引入注意力机制的情况下的学习表现，我们在相同的数据集上使用传统 RNN 解码器结构进行了训练，绘制了其损失变化曲线并采样了若干测试句子进行翻译输出评估。

如图 1 所示，模型训练初期损失值约为 2.6，随后呈现较为平稳的下降趋势，并在 100 个 epoch 内逐渐收敛至约 0.3。整体曲线平滑、无震荡现象，表明模型在训练过程中能够稳定地优化参数，具备一定的收敛能力。

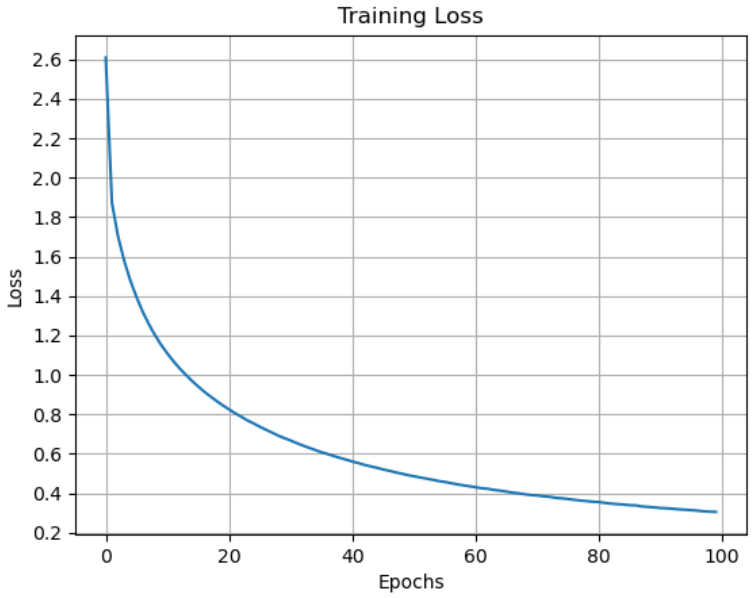



图 1: 训练过程中损失值的变化曲线（RNN 解码器）

尽管损失值最终达到较低水平，我们对模型翻译结果进行了定性评估，部分样例如下所示：

- **输入句（法语）**：*tu ne vas pas mourir ici*
  **参考翻译**: you are not going to die here
  **模型输出**: you re not going to want to miss this
- **输入句（法语）**：*il est bucheron*
  **参考翻译**: he s a lumberjack
  **模型输出**: he s a jailbird on the lam
- **输入句（法语）**：*je remplis un questionnaire*
  **参考翻译**: i am filling in a questionnaire
  **模型输出**: i am a member of the sales department

从上述样例可以看出，尽管模型训练损失较低，其实际生成的翻译在语义对齐方面仍存在明显偏差。一方面，模型缺乏对源语言序列中关键部分的显式关注机制，导致输出结果往往脱离原文语义或结构；另一方面，长句或语义复杂的句子会加剧编码器信息压缩的困难，使得解码器难以恢复完整语义。

这些现象说明，单纯依赖固定长度的编码向量限制了模型的表达能力，尤其在句子长度增加或结构复杂时，传统 RNN 解码器难以捕捉到完整的上下文信息。因此，在后续实验中引入注意力机制，将成为提升模型对齐能力和翻译质量的关键手段。

# 3 基于注意力机制的 Seq2Seq 模型

在经典的 Seq2Seq 模型中，编码器将整个输入序列压缩为一个固定长度的上下文向量作为解码器的初始状态。然而，这种方式在处理长序列时往往面临信息瓶颈问题，导致生成效果下降。为克服这一缺陷，引入注意力机制（Attention）成为解决该问题的有效手段。带注意力机制的解码器能够在生成每一个目标词时，动态地对源序列的不同位置赋予不同的关注权重，从而更加精准地捕捉输入序列中的关键信息。

## 3.1 注意力机制

本模型采用 Bahdanau Attention（Additive Attention）机制，其核心思想是在解码每一个时间步 $t$ 时，使用当前解码器的隐状态（query）与编码器所有时间步的输出（keys）进行匹配计算注意力得分，并根据得分加权聚合编码器输出，形成上下文向量（context）。上下文向量与当前时间步的词嵌入共同作为 GRU 的输入，以输出当前时间步的解码结果。该机制通过学习对齐分布，显著提升了解码器的表达能力。

## 3.2 带注意力的解码器结构

带注意力的解码器由如下几个模块组成：

- **词嵌入层（Embedding Layer）**：将目标词索引转换为稠密向量，维度为 $(N, 1, H)$。
- **Bahdanau 注意力层**：接受当前解码器隐状态（query）与编码器输出序列（keys），计算注意力得分并归一化生成注意力权重 $\alpha$，用于生成上下文向量。
- **上下文拼接与 GRU 层**：将嵌入向量与上下文向量按通道维拼接，作为 GRU 的输入。GRU 计算当前时间步的隐藏状态。
- **输出层（Linear Projection）**：将 GRU 输出映射到词汇表大小维度，用于预测当前时间步的输出词。

## 3.3 训练过程中维度变化分析

在每个时间步 $t$，该解码器的前向传播过程如下：

1. 当前输入 token 的索引 $y_{t-1}$ 被嵌入为向量：$(N, 1) \to (N, 1, H)$。
2. 将前一隐藏状态 $h_{t-1}$ 与编码器输出 $E = \{e_1, \ldots, e_L\} \in R^{N \times L \times H}$ 送入注意力模块，计算注意力分布 $\alpha_t \in R^{N \times L}$ 并生成上下文向量 $c_t \in R^{N \times 1 \times H}$。

3. 拼接嵌入向量与上下文：$(N,1,H)\oplus(N,1,H)\to(N,1,2H)$，传入 GRU 得到当前输出 $o_t$ 和隐藏状态 $h_t$。

4. 输出通过线性层投影为 logits：$(N,1,H)\to(N,1,V)$，$V$ 为目标词汇表大小。

最终，每一个时间步输出一个预测分布以及注意力权重序列 $\alpha_t$，这些权重可视化后具有良好的可解释性，显示模型在生成当前词时关注了输入句子的哪些部分。

### 3.4 实验结果及分析

为了更直观地展示带注意力机制的 Seq2Seq 模型（AttnDecoderRNN）的训练效果与模型行为，我们分别绘制了训练过程中的损失变化曲线以及多个样本对应的注意力权重可视化图。

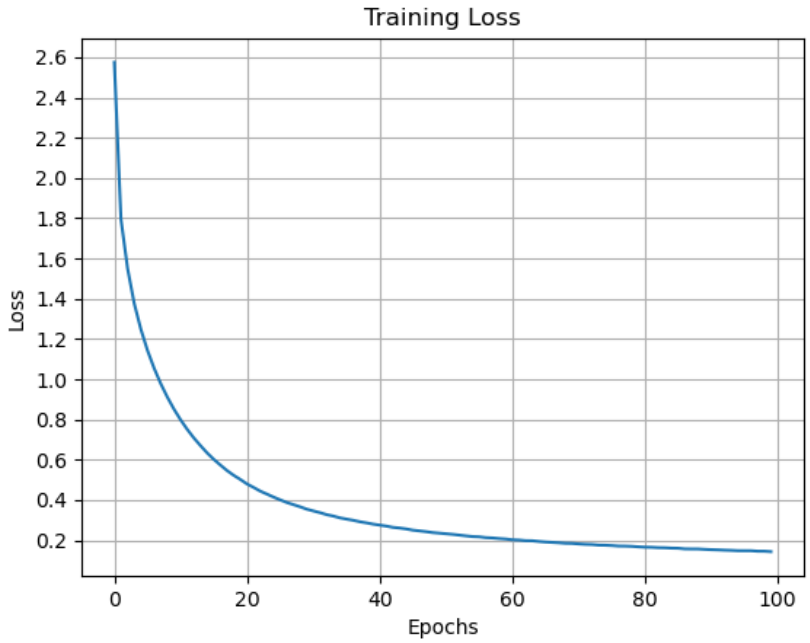
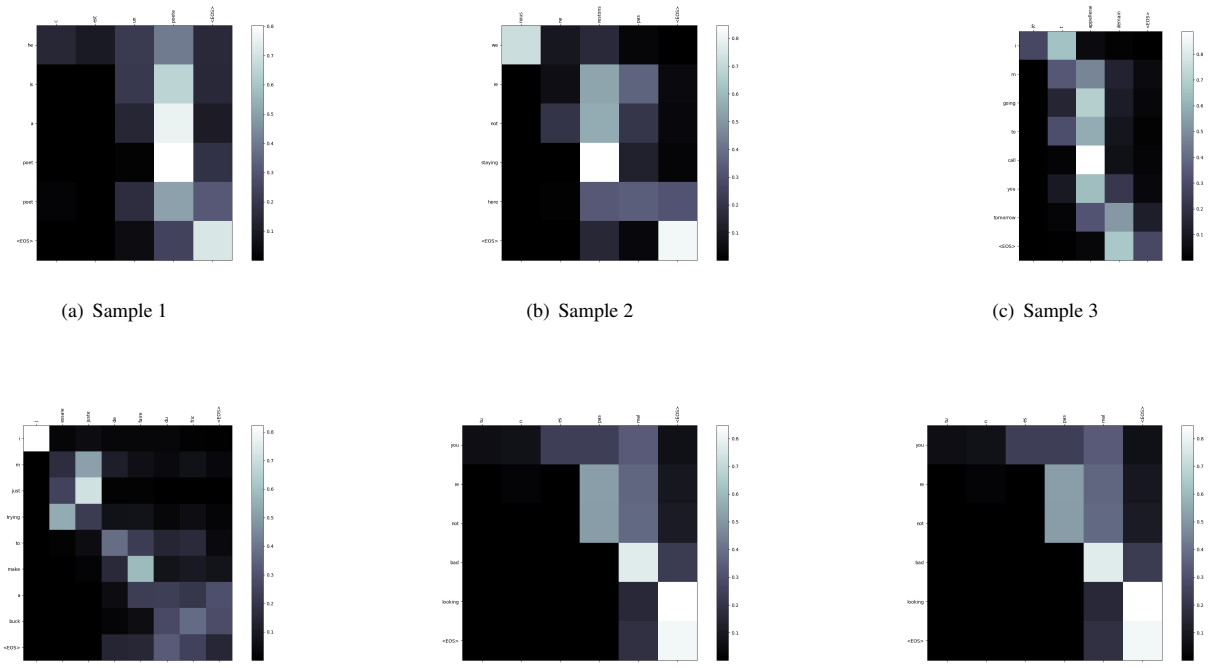



图 2: 训练过程中损失值的变化曲线（融入 attention 的解码器）

如图 2 所示，模型训练初期损失值较高，约在 2.5 左右，但随着训练轮数的增加，损失迅速下降，20 个 epoch 内已降至 1.0 以下，并在 50 epoch 后趋于稳定，最终在 100 epoch 左右收敛至 0.3 附近。该结果表明，模型在训练过程中表现出良好的收敛性与稳定性，且没有出现明显的震荡或过拟合迹象，说明在当前训练集和参数设置下，模型的优化过程是有效且鲁棒的。

为了进一步分析模型的内部机制，我们对测试集中部分翻译样本的注意力分布进行了可视化，结果如图 3 所示。每幅子图展示了一个翻译样本的注意力矩阵，其中横轴为源语言输入词（包含结束标记 `<EOS>`），纵轴为模型生成的目标语言词。



(a) Sample 1 (b) Sample 2 (c) Sample 3

(d) Sample 4 (e) Sample 5 (f) Sample 6

图 3: 注意力权重样例展示

从图中可以观察到，大多数样本的注意力热图呈现出较强的对角线结构，即模型在生成每个目标词时，能够自动对齐到对应的源语言词。这种对齐能力是传统 Seq2Seq 模型所不具备的，充分说明了注意力机制在增强模型表示能力方面的重要性。部分样本（如 Sample 2 和 Sample 5）中，注意力分布具有明显的偏移或多重聚焦现象，表明模型在生成某些目标词时，不仅依赖单一源词，还融合了多个上下文位置的信息，这也反映了注意力机制对长距离依赖建模的潜力。

总体而言，引入注意力机制后的 Seq2Seq 模型在训练效率、收敛表现以及语义对齐等方面均得到了显著提升，尤其适用于需要处理变长输入、存在信息对齐需求的自然语言处理任务，如翻译、对话生成与摘要生成等。

# 4 总结

本实验基于经典的 Seq2Seq 框架，设计并实现了加入注意力机制的编码器-解码器模型，并在英法翻译任务上进行了训练与可视化分析。通过与无注意力机制的基线模型进行对比，实验验证了注意力机制在建模复杂语言结构中的有效性和优越性。

从训练损失的角度来看，加入注意力机制后模型的收敛速度明显加快，训练初期即可快速降低损失，并在后期保持稳定，有效缓解了长序列带来的梯度消失问题。这说明注意力机制提升

了模型对关键信息的聚焦能力，从而增强了整体的表达与生成能力。

更重要的是，通过可视化多个样本的注意力权重图可以发现，模型能够自动学习输入序列与输出序列之间的词对齐关系。在大多数情况下，目标语言中的词能够准确地对齐到对应的源语言词上，注意力矩阵呈现出近似对角线结构，这种可解释性是传统 Seq2Seq 模型难以实现的。

此外，在面对句法较复杂或词义需要上下文判断的场景时，注意力机制能够融合多个输入位置的信息，使得模型生成的输出更加自然、连贯。这一优势使得引入注意力机制的模型在机器翻译、文本摘要、问答系统等多种自然语言处理任务中展现出广泛的适应性与实用价值。

综上所述，注意力机制不仅显著提升了模型在训练效率和泛化能力方面的表现，同时赋予了神经网络模型一定程度的可解释性，进一步拓展了神经机器翻译等应用的实际边界。后续工作中可以考虑引入更高级的注意力结构，进一步提升模型性能。