应用统计 2：时间序列 之 第二次作业

GRU 模型

221870091 蔡如意

指导老师：刘帆

南京大学 - 工程管理学院

目录

第一部分

GRU 的提出背景

背景 1：RNN 的局限性

- RNN 的核心原理

$$h_t = f_W(h_{t-1}, x_t)$$

  - 即对于每个时间步，隐藏层的净输入 $z_t = Uh_{t-1} + Wx_t + b$
  - 隐藏层的状态为 $h_t = f(z_t)$，其中 f 为非线性激活函数。

- 更新参数：时间反向传播 (BPTT)
  - 损失函数为 L_t，定义误差项 $\sigma_{t,k} = \frac{\partial L_t}{\partial z_k}$ 为第 t 时刻损失对第 k 时刻隐藏神经层净输入的导数，即

$$\sigma_{t,k} = \frac{\partial L_t}{\partial z_k} = \frac{\partial h_k}{\partial z_k}\frac{\partial z_{k+1}}{\partial h_k}\frac{\partial L_t}{\partial z_{k+1}} = diag(f'(z_k))U^T\sigma_{t,k+1}$$

  - 整个序列的损失函数 L 关于参数 U，权重 W 和偏置 b 的梯度分别为

$$\frac{\partial L}{\partial U} = \sum_{t=1}^{T}\sum_{k=1}^{t}\sigma_{t,k}h_{k-1}^T \quad \frac{\partial L}{\partial W} = \sum_{t=1}^{T}\sum_{k=1}^{t}\sigma_{t,k}x_k^T \quad \frac{\partial L}{\partial b} = \sum_{t=1}^{T}\sum_{k=1}^{t}\sigma_{t,k}$$

背景 1：RNN 的局限性

- 梯度爆炸或消失：RNN 通过时间反向传播（BPTT）更新参数时，梯度会随着时间步呈指数级衰减或爆炸，导致远距离依赖难以捕捉。
  - 梯度爆炸：当 $diag(f'(z_k))U^T > 1$ 时，如果时间间隔过大，$\sigma_{t,k}$ 会趋向无穷，产生梯度爆炸问题
  - 梯度消失：当 $diag(f'(z_k))U^T < 1$ 时，如果时间间隔过大，$\sigma_{t,k}$ 会趋向 0，产生梯度消失问题
- 记忆容量有限：RNN 的隐藏状态（Hidden State）需同时承担“记忆历史信息”和“生成当前输出”的双重任务，难以长期保留关键信息。

背景 2：LSTM 的局限性

- LSTM 采用两大机制解决 RNN 的缺点
  - 梯度消失或爆炸的问题：采用门控机制（输入门、遗忘门、输出门）解决
    - 遗忘门：$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
    - 输入门：$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$
    - 输出门：$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$
  - 短期记忆覆盖长期记忆的问题：采用记忆单元（Cell State）来保存长期记忆
    - 记忆的更新：$\tilde{c}_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c) \quad c_t = f_t * c_{t-1} + i_t * \tilde{c}_t$

背景 2：LSTM 的局限性

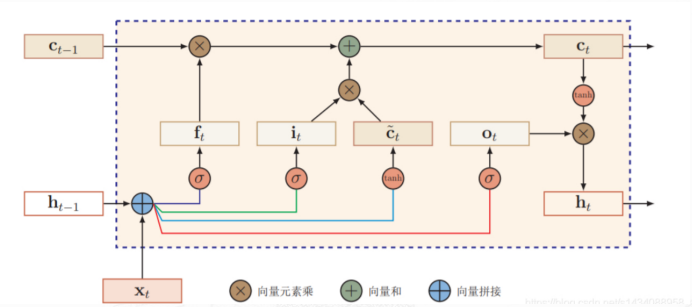

图 1.1: LSTM 原理图

- LSTM 解决了梯度问题，但结构复杂。
  - 参数量大：LSTM 包含 3 个门控和 1 个记忆单元，计算复杂度高。
  - 训练效率低：复杂的结构导致训练速度较慢，尤其在长序列场景下。

目录

第二部分

GRU 的原理

GRU 的核心

- GRU（Gated Recurrent Unit）由 Cho 等人在 2014 年提出，目标是简化 LSTM 结构，同时保留其门控机制的优势：
  - 合并门控：将 LSTM 的输入门和遗忘门合并为更新门，减少参数数量。
  - 统一隐藏状态：取消记忆单元，直接通过隐藏状态传递信息，简化计算流程。
- GRU 的核心是两个门控机制：更新门（Update Gate）和重置门（Reset Gate），通过动态控制信息流动解决长程依赖问题。

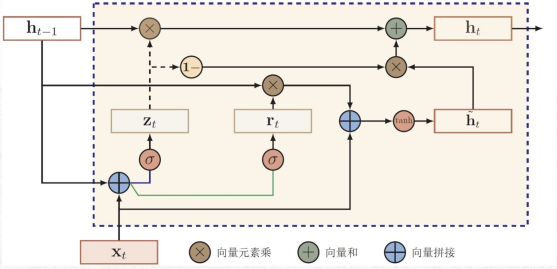

图 2.1: GRU 原理图

GRU 的原理：更新门

- **更新门**：决定当前时刻隐藏状态应保留多少历史信息，并融合多少新信息。

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

- 其中，
  - x_t 为第 t 个时间步的输入向量。
  - W_t 为权重矩阵。
  - h_{t-1} 为上一时刻的隐藏状态，即 $t-1$ 时间步的信息。
  - σ 为 Sigmoid 函数，将输出压缩到 0 - 1 之间。
- z_t：更新门的输出（取值 0-1），控制历史信息的保留比例。

GRU 原理：重置门

- **重置门**：决定是否忽略历史信息以生成新的候选状态，即到底遗忘过去的多少信息。

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

- 其中，
  - x_t 为第 t 个时间步的输入向量。
  - W_r 为权重矩阵。
  - h_{t-1} 为上一时刻的隐藏状态，即 $t-1$ 时间步的信息。
  - σ 为 Sigmoid 函数，将输出压缩到 0-1 之间。
- r_t：重置门的输出（取值 0 - 1），接近 0 时表示忽略历史信息。

GRU 原理：候选隐藏状态和隐藏状态更新

- 候选隐藏状态：结合重置门和历史信息，生成当前时刻的候选状态

$$\tilde{h}_t = tanh(W \cdot [r_t * h_{t-1}, x_t])$$

  - $r_t * h_{t-1}$: 重置门控制历史信息的过滤，若 $r_t \approx 0$ 则丢弃历史信息，仅依赖当前输入。

- 隐藏状态更新：通过更新门融合历史状态和候选状态

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_{t-1}$$

  - $z_t \approx 1$：隐藏状态主要由候选状态更新（关注新信息）。
  - $z_t \approx 0$：保留大部分历史信息（忽略当前输入）。

目录

第三部分

GRU 的应用

GRU 的应用场景

- 自然语言处理（NLP）
  - 机器翻译：捕捉源语言和目标语言的上下文依赖（如 Google 的早期翻译模型）。
  - 文本生成：生成连贯的对话或文章（如聊天机器人、自动摘要）。
  - 情感分析：分析长文本中的情感倾向。
- 时间序列预测
  - 股票价格预测：基于历史价格序列预测未来趋势。
  - 气象预测：处理时间相关的气象数据（如温度、湿度序列）。
- 语音识别
  - 声学建模：将语音信号映射为文本序列，捕捉语音中的时序特征。
- 推荐系统
  - 用户行为建模：根据用户历史行为序列（点击、购买记录）预测兴趣。

GRU 的优缺点

- 优势：
  - 简洁高效：GRU 的结构相对简单，参数较少，训练速度快。
  - 解决梯度问题：通过引入门机制，GRU 有效地解决了传统 RNN 中的梯度消失和爆炸问题，从而能够更好地捕捉序列数据中的长期依赖关系。
  - 适应性强：可以用于处理各种类型的序列数据，包括文本、音频、图像等。
- 限制：
  - 对于非常长的序列，GRU 可能无法完全捕捉所有的长期依赖关系。因为尽管门机制帮助控制信息的传递，但在非常长的序列中信息的传递仍会受到一定的限制。
  - GRU 难以显式建模序列中的层次结构。如，在自然语言处理任务中，词语的含义可能取决于它在句子中的位置，而句子的含义可能取决于它在段落中的位置。这种层次结构是 GRU 难以处理的。

LSTM VS GRU

对比维度	LSTM	GRU
门控机制	3 个门：输入门、遗忘门、输出门	2 个门：更新门、重置门
记忆单元	独立细胞状态（Cell State）	无独立细胞状态，通过更新门和重置门联合控制
参数量	较多（多一个门控和细胞状态）	较少（参数更精简）
计算复杂度	较高（需维护细胞状态）	较低（合并门控和状态）
训练速度	较慢（参数多）	较快（参数少）
长依赖捕捉	更强（显式控制记忆遗忘）	稍弱（隐式记忆更新）
适用场景	超长序列、复杂时序依赖（如机器翻译）	中等序列、实时性要求高（如语音识别）
梯度消失问题	缓解（通过细胞状态）	缓解（通过更新门）
主流框架实现	广泛支持	广泛支持

表 3.1: GRU 和 LSTM 对比

谢谢倾听！