13 长期依赖、门控与编码器—解码器

选择性更新、编码解码范式与条件生成

学习目标

门控讨论状态内部怎样更新；编码器—解码器讨论输入怎样通过表示影响输出。本讲用GRU Seq2Seq把两者结合。

- 解释长期信息保留与反向学习信号的联系和区别。
- 跟踪GRU更新，比较LSTM的状态分工与双向使用条件。
- 解释编码器、表示接口与解码器的职责，区分范式与具体实现。
- 跟踪固定上下文GRU Seq2Seq的两条递推及其限制。
- 说明目标移位、有效损失怎样联合训练编码与解码。
- 区分真实前缀训练与模型反馈生成，并跟踪给定短例。

1 长期依赖与门控更新

从信息保留问题建立选择性状态更新

一条信息需要保留多久

- 在一个序列识别任务中，模型需在末步报告首标记是A还是B；中间输入不改变正确答案
  - 例如，输入“A、3、7、2”时末步应报告A；输入“B、3、7、2”时应报告B
- **承载首标记区别的状态分量需要跨步保留**；普通RNN持续更新可能改写这些信息
- 需要学习哪些分量保留、哪些更新；后期损失还需通过早期计算训练共享参数

序列识别任务：末步报告首标记，箭头表示时间顺序

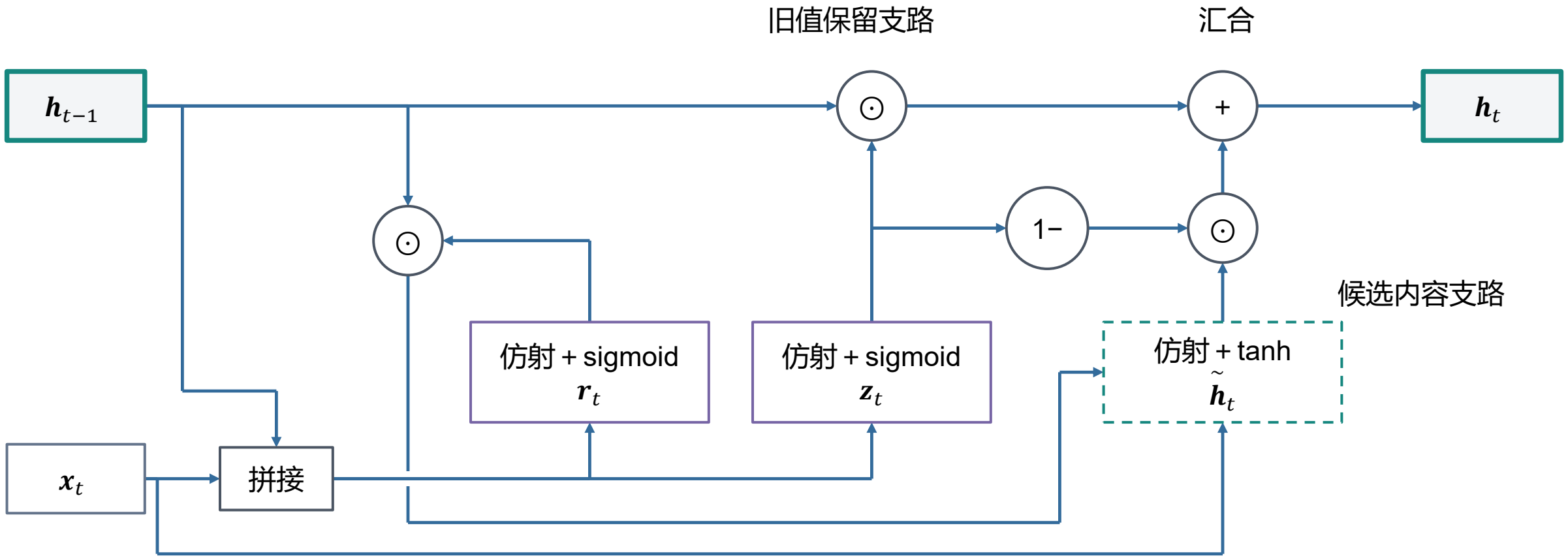


门是按输入计算的连续系数

- **门控**用当前输入和旧状态计算系数，再控制相应信息的保留或写入
- sigmoid函数 $\sigma(a) = 1/(1 + \exp(-a))$ 将有限实数映射到0与1之间
- **门向量可对不同状态分量使用不同系数**，而不是只给整个状态一个开关
- $\odot$ 表示逐分量相乘，例如 $[0.2,0.8] \odot [10,10] = [2,8]$
- 门值来自可学习参数的计算，不需要人工为每个时刻指定

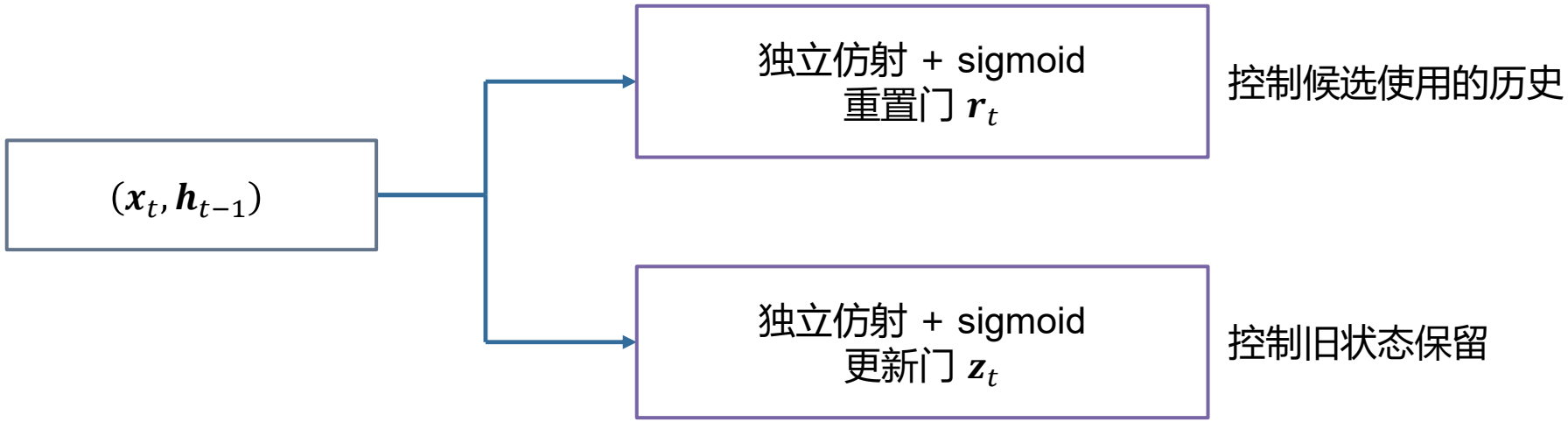
GRU的门控更新结构

- **门控循环单元**（GRU）有两条支路：旧值直接保留，候选准备新内容
- **重置门** r_t 调节候选使用的历史；**更新门** z_t 控制两支路的混合比例



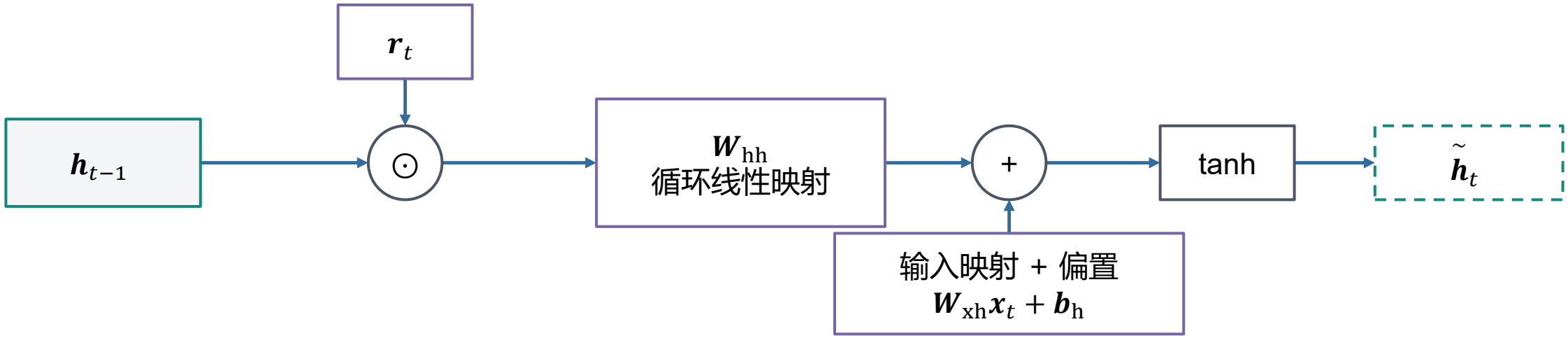
由输入与旧状态计算两种门

- 输入 $\boldsymbol{x}_t \in \mathbb{R}^d$，旧状态 $\boldsymbol{h}_{t-1} \in \mathbb{R}^m$，均为列向量
- 重置门：$\boldsymbol{r}_t = \sigma(\boldsymbol{W}_{\mathrm{xr}}\boldsymbol{x}_t + \boldsymbol{W}_{\mathrm{hr}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\mathrm{r}})$
- 更新门：$\boldsymbol{z}_t = \sigma(\boldsymbol{W}_{\mathrm{xz}}\boldsymbol{x}_t + \boldsymbol{W}_{\mathrm{hz}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\mathrm{z}})$
- 输入权重为 $m\times d$，循环权重为 $m\times m$，偏置和门为 m 维；两门参数独立



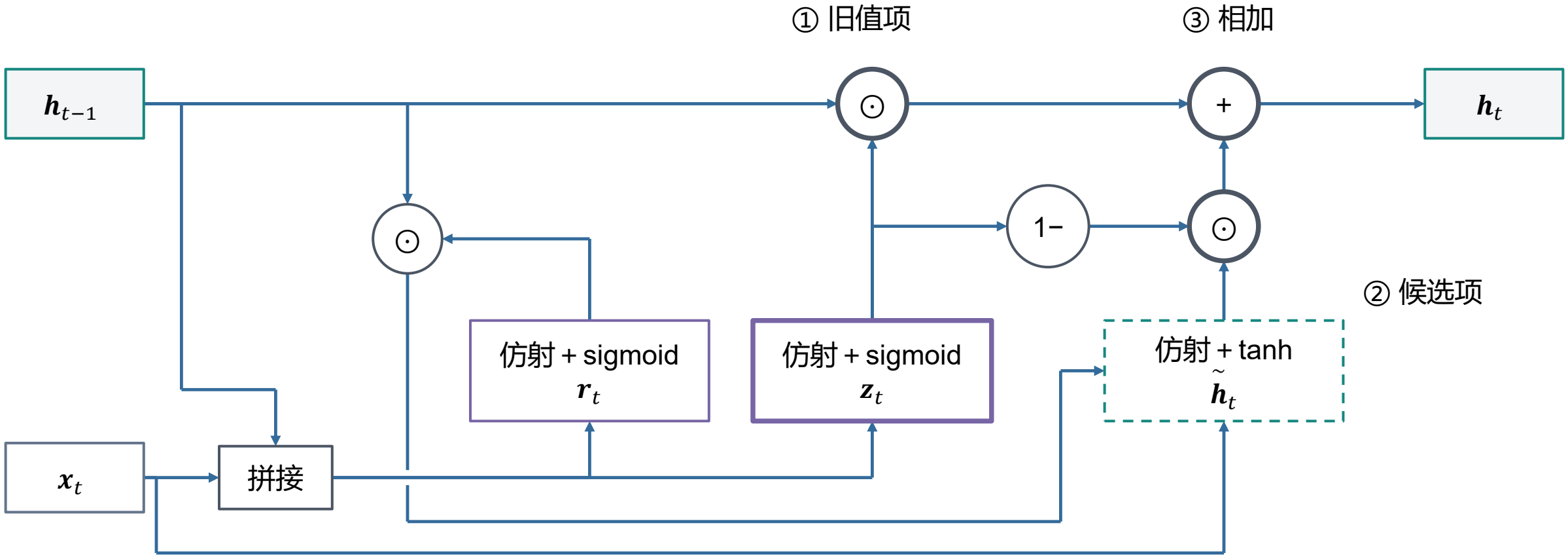
重置后的历史参与候选计算

- 按本讲D2L约定，先重置旧状态，再作循环线性映射：
  - $\tilde{\boldsymbol{h}}_t = \tanh(\boldsymbol{W}_{\mathrm{xh}}\boldsymbol{x}_t + \boldsymbol{W}_{\mathrm{hh}}(\boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}) + \boldsymbol{b}_{\mathrm{h}})$
- 重置门分量接近0时，对应旧分量对候选的影响减弱；**候选还需经过混合才能成为实际状态**
- PyTorch把重置门放在线性映射之后，复现时须核对次序，不能直接混用参数



旧状态与候选的逐分量混合

- 本讲更新为 $\boldsymbol{h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t}$
- **①旧值加权、②候选加权、③相加得到实际状态**；z_t 越接近1越偏向旧值



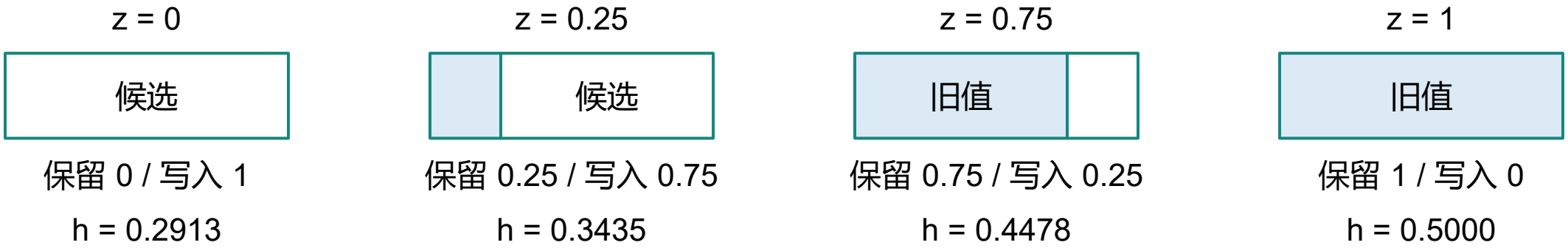
一个完整标量GRU更新

计算对象	本例计算	结果
重置门 r	$\sigma(x-2h_{t-1})=\sigma(0)$	0.5
更新门 z	$\sigma((\ln 3)x)=\sigma(\ln 3)$	0.75
候选	$\tanh(0.2x+0.4rh_{t-1})$	0.2913
实际新状态	$zh_{t-1}+(1-z)\tilde{h}_t$	0.4478

- 给定 $x=1, h_{t-1}=0.5$，三组仿射参数由表内公式完整规定
- 候选预激活为 $0.2+0.4\times0.5\times0.5=0.3$，本例没有训练过程

保留系数与状态混合

- 固定旧状态0.5、候选0.2913，只改变更新门：$h = z \times 0.5 + (1 - z) \times 0.2913$
- 条长表示系数，0和1为理想极限；真实模型中门与候选都会随输入改变

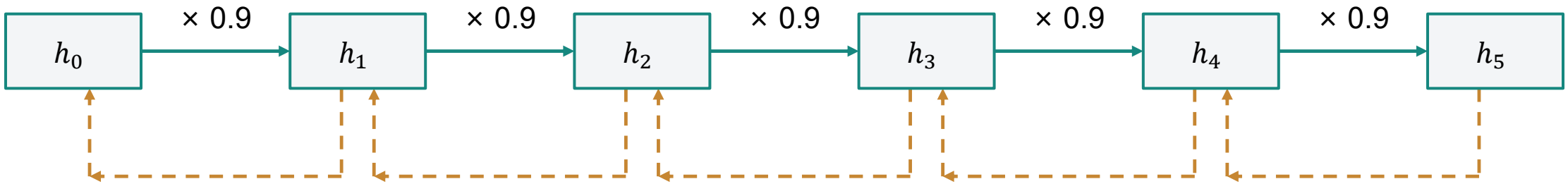


淡蓝段：旧状态系数 z；白色段：候选系数 1-z（两者之和为1）

直接保留支路与完整梯度

- **前向保留**：更新门接近1时，有用旧分量可较少改写地传到后续时刻
- **反向支路**：标量例中，所选直接支路的五步系数乘积为 $0.9^5 = 0.59049$
- **完整梯度**还含门与候选对旧状态的依赖；门控不保证消除梯度消失

实线向右：前向状态传递



虚线箭头：沿选定保留支路向左回传

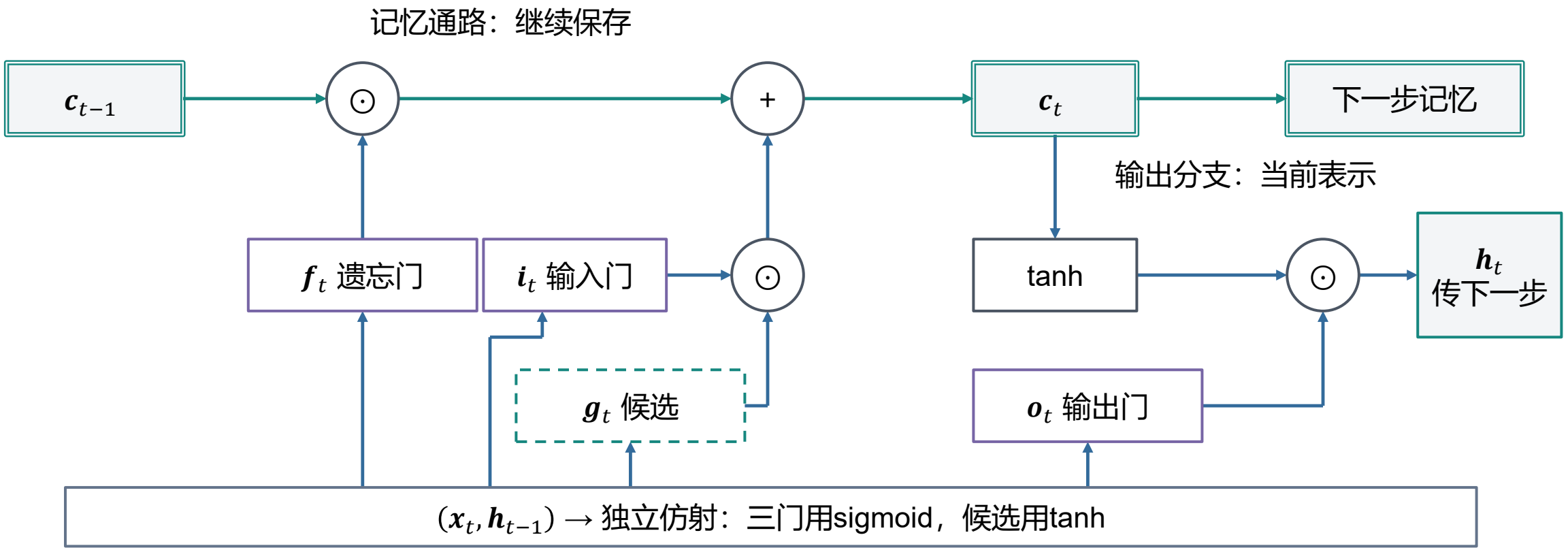
各步的门与候选还依赖旧状态：完整梯度需合计所有依赖路径

2 LSTM的记忆与输出

从同一隐藏状态承担保存与表示，比较记忆更新和当前输出的分工

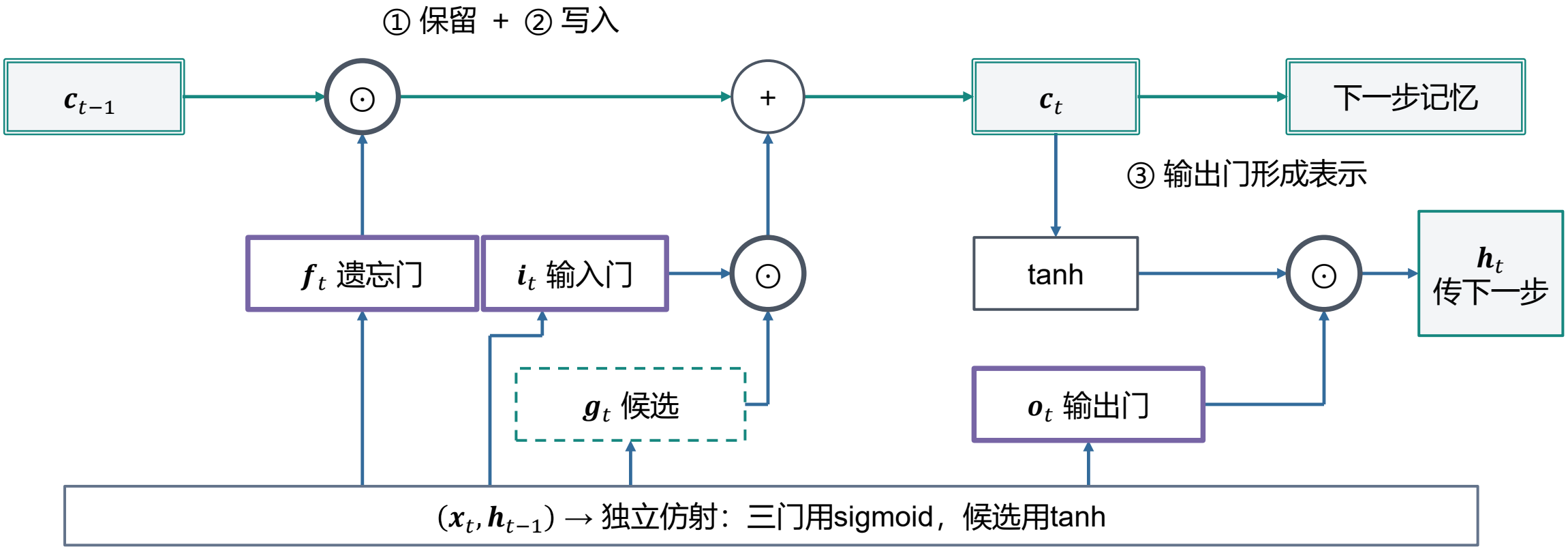
LSTM的双状态结构

- **长短期记忆网络**（LSTM）分别维护 m 维记忆 $\boldsymbol{c}_t$ 与隐藏输出 $\boldsymbol{h}_t$
- **记忆通路**：$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \boldsymbol{g}_t$，继续传到下一步
- **输出分支**：$\boldsymbol{h}_t = \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t)$，再接任务输出头；保存的信息不必全部当前输出



记忆更新与对外输出

- **①保留** $f_t \odot c_{t-1}$ 与 **②写入** $i_t \odot g_t$ 相加，形成新记忆 c_t
- **③输出**：o_t 调节 $\tanh(c_t)$ 形成 h_t，不直接清空记忆
- 遗忘门与输入门不必互补，记忆不限于 $[-1,1]$；h_t 与下一步输入计算新门和候选



给定门值的一次记忆更新

对照	内部记忆 c_t	隐藏输出 h_t
给定	$c_{t-1}=1.2, g_t=-0.4;\ f_t=0.8, i_t=0.25$	$o_t=0.6$
本步结果	0.86	约0.4178
当前职责	继续保存，供下一步更新	提供当前表示，也传下一步

- 新记忆 $c_t = 0.8\times1.2 + 0.25\times(-0.4) = 0.86$
- 隐藏输出 $h_t = 0.6\tanh(0.86) \approx 0.4178$：**保存的信息与当前对外表示可以不同**

GRU与LSTM的状态组织对照

比较项	本讲GRU	本讲LSTM
递推状态	隐藏状态 $\boldsymbol{h}$	记忆 $\boldsymbol{c}$ 和隐藏输出 $\boldsymbol{h}$
门	重置、更新	遗忘、输入、输出
保留与写入	$\boldsymbol{z}$ 与 $\mathbf{1}-\boldsymbol{z}$ 互补	$\boldsymbol{f}$ 与 $\boldsymbol{i}$ 分别计算
对外表示	混合后的 $\boldsymbol{h}$	由记忆经tanh和输出门形成 $\boldsymbol{h}$

- LSTM的保留与写入系数不要求相加为1，记忆更新的限制与GRU不同
- 结构差别提供不同选择，效果比较仍须固定任务、训练和评价条件

3 编码器—解码器与可见信息

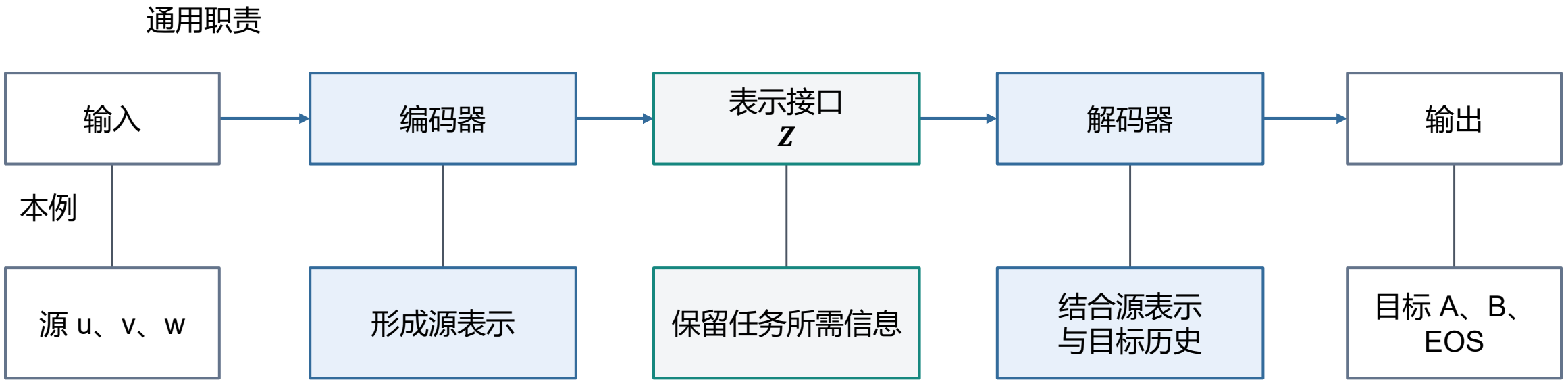
源与目标如何联系？本节从状态更新转向编码与解码的职责。

源序列与目标序列需要分别组织

- 下一词元预测沿同一序列继续；翻译或摘要则需要在给定源序列下产生另一条序列。
- **源与目标的长度、词表和词序可以不同**，目标位置通常不与源位置逐项对应。
  - 例如，“我明天去北京”可译为“I will go to Beijing tomorrow”；时间表达从句中移到句末。
- **序列到序列**（Seq2Seq）模型学习源序列到目标序列的映射。
- 本讲沿用符号样本：源为u、v、w，目标内容为A、B；模型还要决定何时结束。

编码与解码是通过表示连接的两种职责

- 源与目标不逐项对齐：**先把源组织成表示，再按目标任务的规则形成输出**
- **编码器**从输入形成表示；**表示接口**决定解码器能够取得哪些输入信息
- **解码器**利用接口组织输出；自回归是本讲接着选择的生成方式，并非范式的必要条件



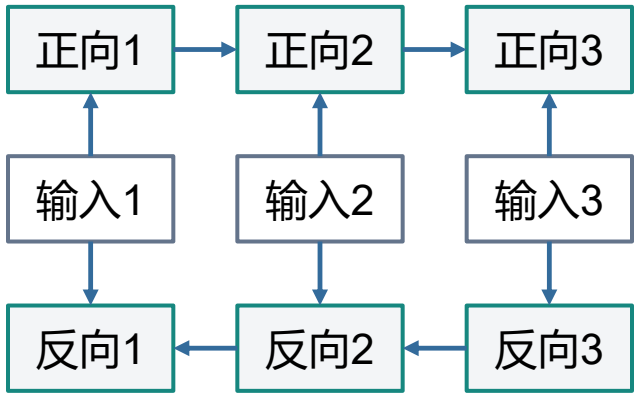
灰竖线表示职责对应；表示可为摘要或多位置特征，输出也可按空间位置组织

序列解码是在源条件下逐步预测

- 编码结果记为 $\boldsymbol{Z} = Enc(a_{1:N})$，其中 $a_{1:N}$ 为长度 N 的源序列。
- **本讲选择自回归输出**：模型以编码表示作为源条件，按已有目标前缀逐步预测：
  - $p(y_{1:J} \mid a_{1:N}) = \prod_{j=1}^{J} p(y_j \mid y_{<j}, \boldsymbol{Z})$
- $y_{<j}$ 是已有目标前缀，J 包含结束标记EOS所在的位置。
- 对本例目标A、B、EOS，三个因子分别对应三个预测位置：
  - $p(A, B, \mathrm{EOS} \mid a_{1:N}) = p(A \mid \boldsymbol{Z})p(B \mid A, \boldsymbol{Z})p(\mathrm{EOS} \mid A, B, \boldsymbol{Z})$
- **源表示固定，目标前缀逐步增长**；首步由BOS启动，公式省略这一固定条件。

同一输入的两个递推方向

- “他吃了苹果”与“苹果发布新手机”：前文或后文可帮助区分同一词元的含义
- **源输入完整给定时，双向循环分别汇集前文和后文**
- 位置 t 把正向状态 $\boldsymbol{h}_t^{\mathrm{f}}$ 与反向状态 $\boldsymbol{h}_t^{\mathrm{b}}$ 拼接；两个方向参数独立
- 每个方向状态为 m 维时，该位置拼接表示为 $2m$ 维



哪些任务能读取后文

场景	是否可使用完整输入的双向表示
整段文本分类	可以，整段输入已知
翻译的源序列编码	可以，源句通常已完整给定
预测尚未到达的传感器数据	不能借用未知后续测量
自回归目标生成	不能读取尚未生成的目标词元

- **“源输入已知”与“目标未来未知”是不同条件**，不能把双向源编码的许可推广到目标生成

同一职责划分可以采用不同实现

实现或任务	编码侧提供什么	解码侧怎样组织输出
固定上下文RNN Seq2Seq	源序列摘要向量	递推状态，逐词产生目标
带注意力的序列模型	一组源位置表示	按当前需要读取，再预测目标
编码器—解码器Transformer	源词元表示序列	结合目标历史和源表示
U-Net语义分割	多尺度特征，跳跃连接保留细节	恢复空间分辨率并预测像素标签

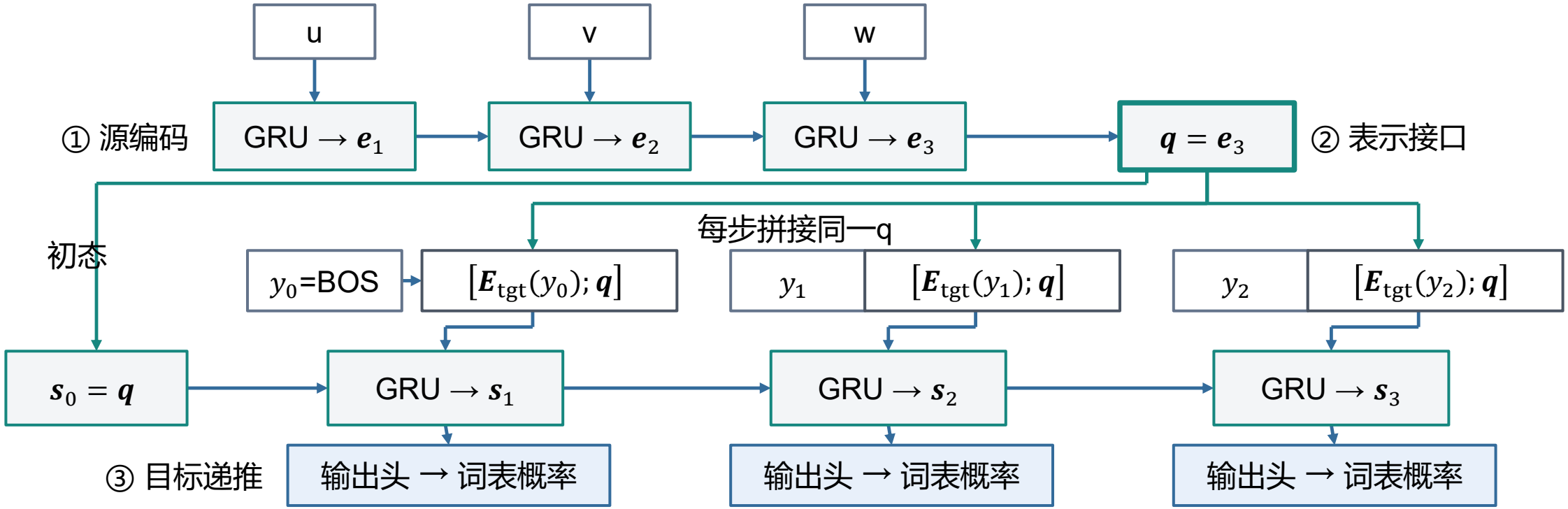
固定向量、逐词生成及RNN递推分别属于具体实现选择；空间恢复与时间生成处理不同的输出对象。

4 固定上下文GRU Seq2Seq

用两条递推落实源条件与目标历史的接口

固定上下文Seq2Seq的双时间轴

- **共同计算：①源编码，②上下文两种用途，③目标递推与词表概率**
- GRU负责状态更新；编码器—解码器负责连接两条序列。两端GRU参数独立



前词元来源　训练：真实目标的前一词元（移位标签）

生成：上一步已选词元（首步共同用BOS）

源序列怎样形成上下文

- 第 i 个源词元为 a_i，源嵌入表给出 $\boldsymbol{x}_i^{\mathrm{src}}$
- 编码递推：$\boldsymbol{e}_i = GRU_{\mathrm{enc}}(\boldsymbol{x}_i^{\mathrm{src}}, \boldsymbol{e}_{i-1})$，初态为零
- 上图中u、v、w依次形成 $\boldsymbol{e}_1, \boldsymbol{e}_2, \boldsymbol{e}_3$，本例取 $\boldsymbol{q} = \boldsymbol{e}_N = \boldsymbol{e}_3$
- $\boldsymbol{q}$ 为 m 维学习表示；有填充时按有效源长度取末状态
- 其他实现可用双向编码器；本讲实例固定采用单向GRU

解码状态结合源条件与目标历史

- **初态决定解码器从什么状态开始**：$\boldsymbol{s}_0 = \boldsymbol{q}$，本例二者均为$m$维。
- **每步拼接使源条件持续参与状态更新**：前词元嵌入与上下文作特征拼接：
  - $\boldsymbol{d}_j = [\boldsymbol{E}_{\text{tgt}}(y_{j-1}); \boldsymbol{q}]$，首步 $y_0 = \text{BOS}$。
- **解码递推**：$\boldsymbol{s}_j = GRU_{\text{dec}}(\boldsymbol{d}_j, \boldsymbol{s}_{j-1})$。
- 输出得分 $\boldsymbol{v}_j = \boldsymbol{W}_{\text{out}}\boldsymbol{s}_j + \boldsymbol{b}_{\text{out}}$，经softmax得到目标词表上的**分布**。

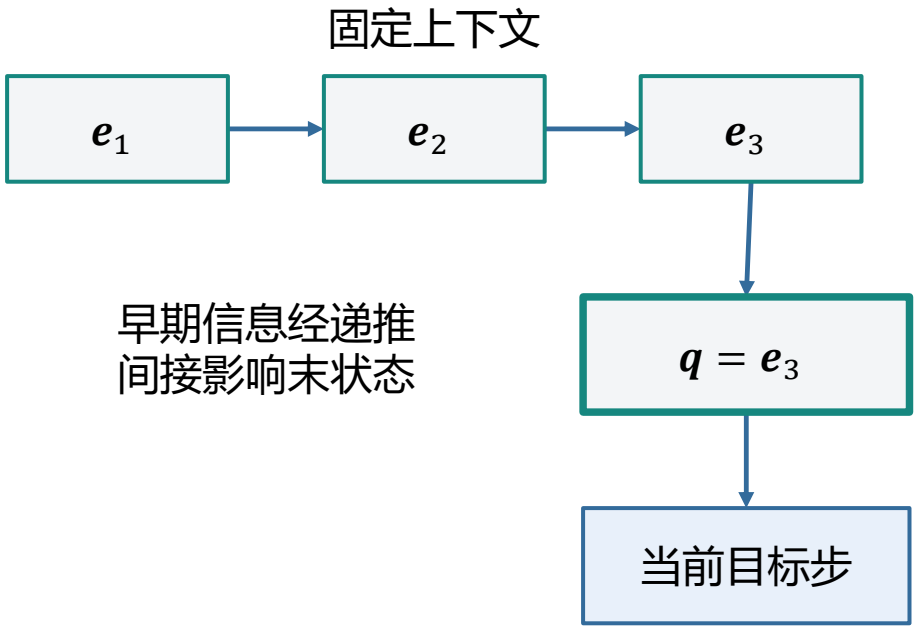
源轴与目标轴具有不同长度

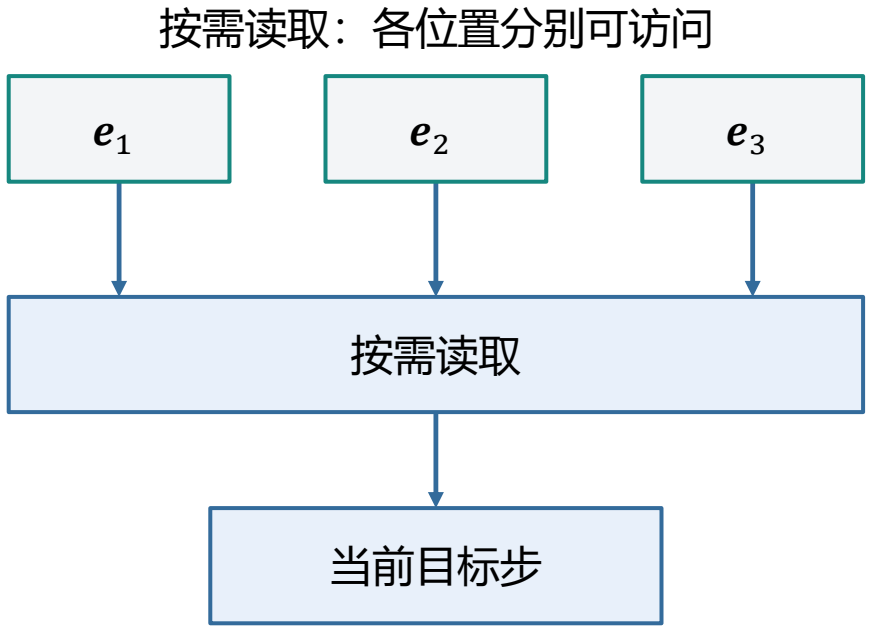
对象	批量存储形状	轴的顺序
源嵌入，源长3	$B\times 3\times d_x$	批量 × 源时间 × 特征
固定上下文	$B\times m$	批量 × 上下文特征
目标嵌入，补齐5步	$B\times 5\times d_y$	批量 × 目标时间 × 特征
每步拼接同一q	$B\times 5\times (d_y+m)$	批量 × 目标时间 × 拼接特征
目标类别得分	$B\times 5\times V_y$	批量 × 目标时间 × 类别

- B 为批量大小；d_x, d_y 为嵌入维数，m 为上下文维数，V_y 为词表大小。
- **源长3、目标补齐5步、有效目标A/B/EOS共3个。**单步对应一个时间切片，补齐位置不计损失。

固定上下文的限制与按需读取

- **无论当前目标步需要哪个源位置，都只能通过同一个q获取源信息**
- 固定摘要限制信息访问，长源细节可能难以保留；并非固定维向量必然丢失信息
- 门控改变保留与写入；按需访问改变可读源表示。**先完成当前模型的训练与生成，按需读取留待下一讲**





下一讲解释怎样依据当前目标需要读取源位置

5 目标移位与联合训练

先完成当前固定上下文模型的训练与生成；按需读取的计算留待下一讲。

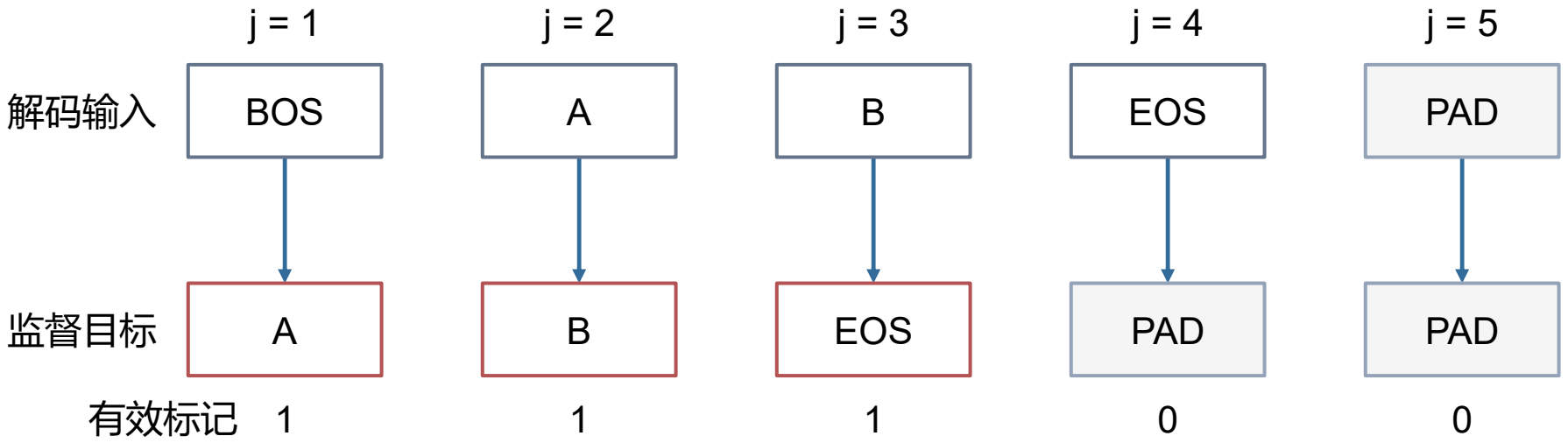
起始、结束与填充有不同职责

标记	本讲中的作用	是否为有效预测目标
BOS	解码第1步的起始输入	不作为本例内容目标
EOS	目标内容结束后的标记	是，模型需要学会结束
PAD	补齐批量数组长度	否，损失排除该位置

- 本例目标词表为A、B、BOS、EOS、PAD，$V_y = 5$
- 起止标记服务于序列接口，不把BOS当作真实内容，也不把EOS当作填充

真实目标与解码输入错开一位

- 目标内容A、B加EOS，补齐至5步；解码输入在前面加入BOS并右移
- 同一列中，**解码器只读先前词元** y_{j-1}，监督目标为 y_j
- EOS参与监督；PAD对应位置的有效标记为0



教师强制使用真实的目标前缀

- **教师强制**（teacher forcing）在训练中把真实的先前目标词元送入解码器
- 对第 j 步，模型学习源序列和真实前缀下的条件概率：
  - $p(y_j \mid a_{1:N}, y_{<j})$
- 例：预测B时，输入真实的A；即使模型第1步本来更偏向别的词元，该训练输入仍按已给标签组织
- **真实的当前目标只用于损失，不直接作为同一步预测的输入**
- 对前面A、B、EOS的三项条件概率乘积取负对数，得到三步交叉熵之和；有效位置平均见下一页。

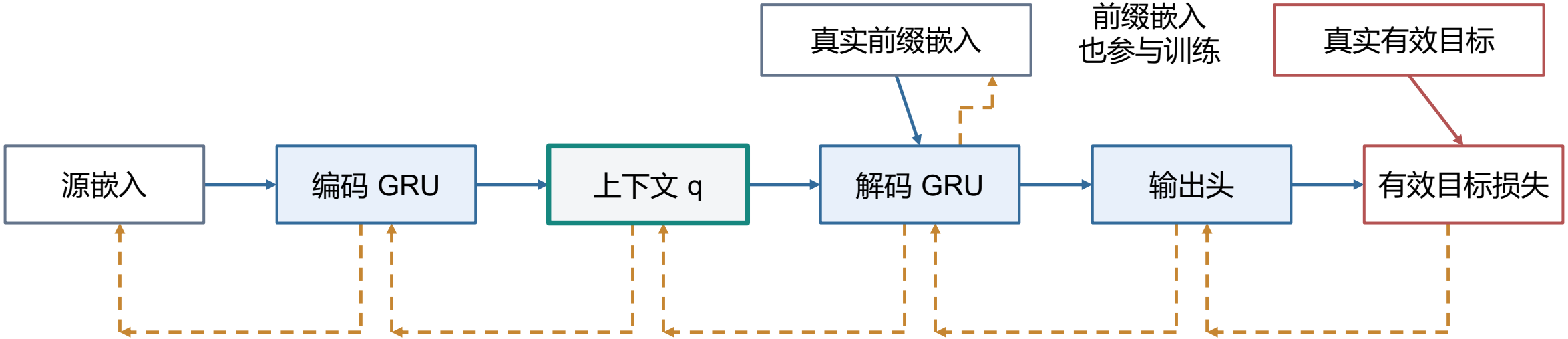
EOS参与监督，PAD不参与平均

真实目标	A	B	EOS	PAD	PAD
有效标记	1	1	1	0	0
给定正确类概率	0.8	0.5	0.9	不计	不计

- 对非空有效位置集合 S，$L = \frac{1}{|S|}\sum_{(n,j)\in S} -\ln p_{n,j,y_{n,j}}$，$n$ 表示批量中的样本
- 本例 $L = (-\ln 0.8 - \ln 0.5 - \ln 0.9)/3 \approx 0.3406$，分母为3而不是5

一次Seq2Seq训练怎样连接两端

- 解码状态依赖上下文 q，q 由编码递推产生，因此**目标损失也依赖编码器参数**
- **反向沿这些依赖回传**，经输出头、解码器和 q；源嵌入与目标前缀嵌入都接收梯度
- 只需目标端有效标签，无需为每个编码状态另给标签；反向后共同更新两端参数



虚线箭头向左：损失经输出头、解码器和上下文传回编码器与嵌入

6 模型反馈与逐步生成

生成缺少真实目标前缀；本节沿用解码接口，逐步反馈模型选词。

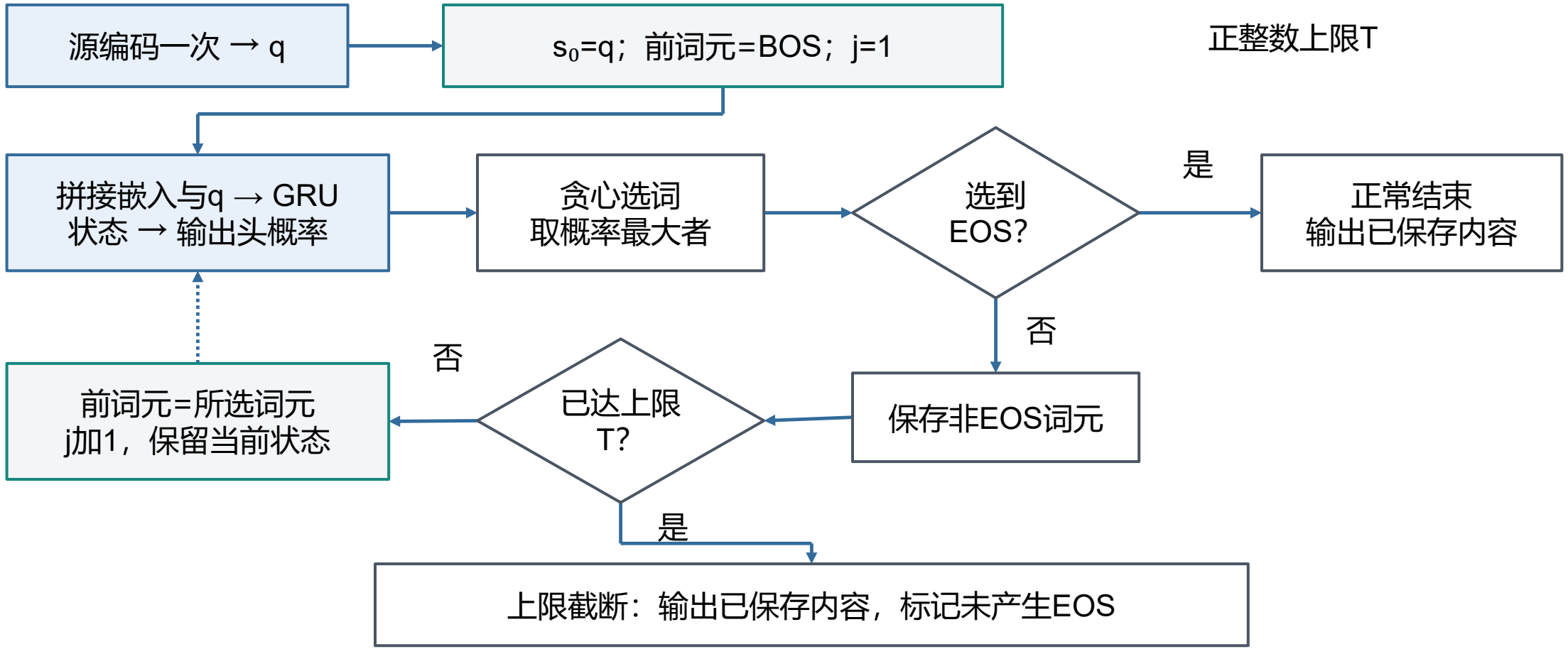
给定条件概率的贪心生成

步	已选内容	p(A)	p(B)	p(EOS)	选择
1	空	0.60	0.24	0.14	A
2	A	0.10	0.70	0.18	B
3	A，B	0.08	0.10	0.80	EOS

- 另给定解码分布，未列出的BOS和PAD概率各为0.01，每行总概率为1
- 从BOS开始，每步选最大概率词元；先得到A、B，随后遇EOS停止

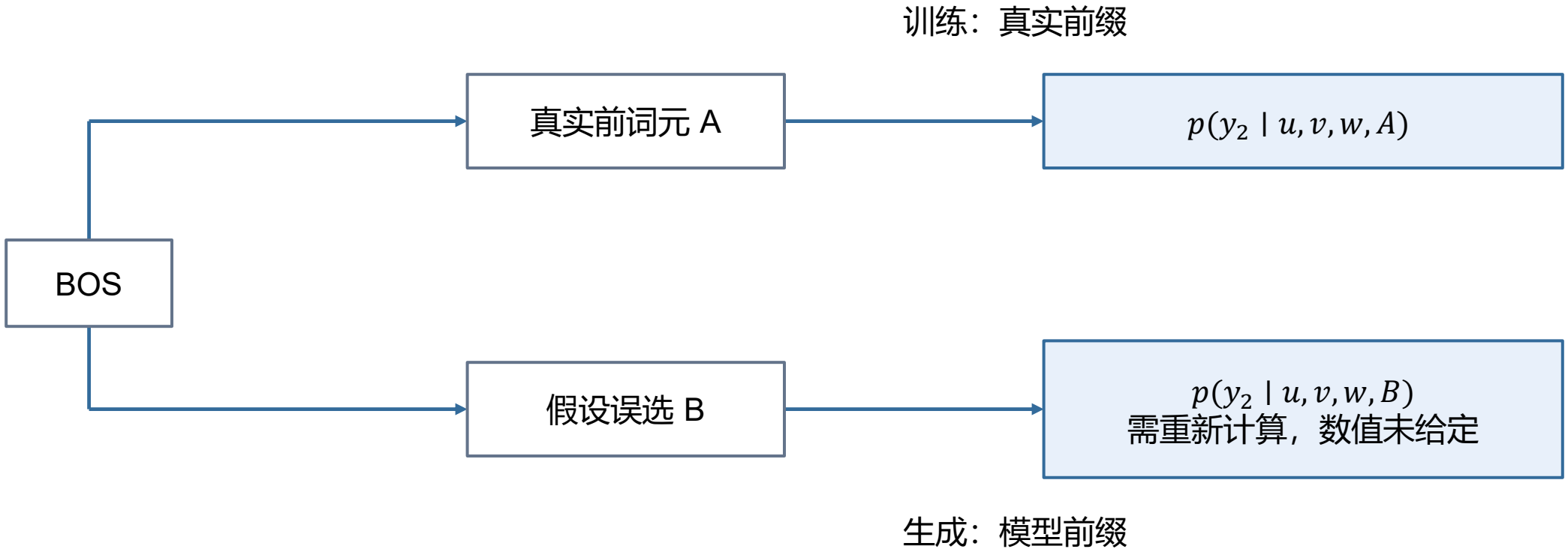
生成时逐步反馈自己的选择

- 同一解码接口：前词元嵌入与q拼接，更新状态，再输出词表概率
- **EOS正常结束与长度上限截断分开；贪心不保证整序列概率最大**



训练与生成改变的是条件前缀

- 训练第2步读真实A；生成若第1步误选B，**第2步就以B为条件重新计算**
- 当前真实标签只用于训练损失；错误前缀后的分布不能沿用前例，数值未给定



本讲总结

- **历史信息如何保留和更新**
  - GRU控制旧值与候选的混合；LSTM区分记忆更新与隐藏输出形成
- **输入如何影响目标生成**
  - 编码器经表示接口影响解码器；GRU Seq2Seq是本讲的具体实现
- 训练用真实前缀和有效损失联合学习；生成反馈所选词元，按EOS或上限停止
- 固定摘要仍限制可访问信息；下一讲从多个源位置按需读取