12 序列建模与循环神经网络

递推状态、共享参数与时间梯度

学习目标

- 解释序列预测为何需要历史，以及固定窗口的范围限制
- 画出递推状态与共享更新规则，跟踪给定短序列的计算
- 连接状态、预测概率与后移目标，区分训练和生成的输入
- 沿时间展开图说明误差传播与共享参数梯度累积
- 区分长期信息保留、梯度传播与分段状态管理的边界

1 序列预测与历史依赖

顺序与可见历史提出表示需求

序列预测问题

阅读一句尚未结束的话时，我们会根据已经读到的内容，判断接下来可能出现的词语。

此前已读历史	当前词元	可能的下一词元
她穿好外套，拉上	了	拉链
演出结束，拉上	了	幕布

- 同一当前词元，历史不同，可能的后续也不同；表中候选非唯一答案
- “学生问老师”与“老师问学生”说明词序影响关系
- 序列由有序元素组成。本讲用下一词元预测贯通模型与学习；连续测量可预测未来数值，序列分类则依据允许的整段输入预测类别

本讲任务：下一词元预测

词表有 V 个词元，第t个词元ID为 $i_t \in \{0, \ldots, V-1\}$。此时已读前缀为 $i_{1:t} = (i_1, \ldots, i_t)$。

参数为 θ 的模型根据这段前缀预测下一词元，给出概率向量 $\boldsymbol{p}_t = (p_{t,0}, \cdots, p_{t,V-1})^{\top}$。其中第j个分量表示

$$p_{t,j} = P_\theta(i_{t+1} = j \mid i_{1:t})$$

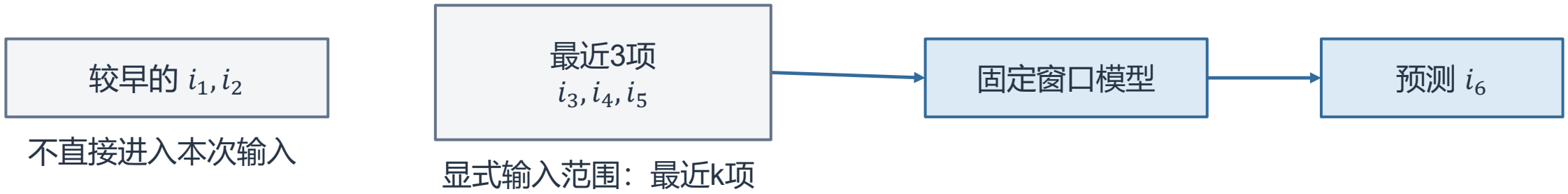
这些概率满足 $p_{t,j} \geq 0$ 和 $\sum_{j=0}^{V-1} p_{t,j} = 1$。

真实的下一词元记为 $y_t = i_{t+1}$，训练用它监督预测；该步预测只使用截至t的前缀。

随着前缀增长，还需要决定怎样组织这些历史输入。

固定窗口与历史范围

- 若把整个已读前缀直接拼接为输入，输入长度会逐步增长；先考虑一种固定输入维度的基础模型，每次取最近 k 项
- 取 $k = 3$：已读 $i_1, \ldots, i_5$ 时，用 i_3, i_4, i_5 预测 i_6；读入真实 i_6 后，用 i_4, i_5, i_6 预测 i_7
- 窗口外元素不直接进入本次预测；扩大固定拼接窗口通常需要调整输入组织
- 相关线索可能出现在不同距离：需要一种能逐步更新历史表示的方案



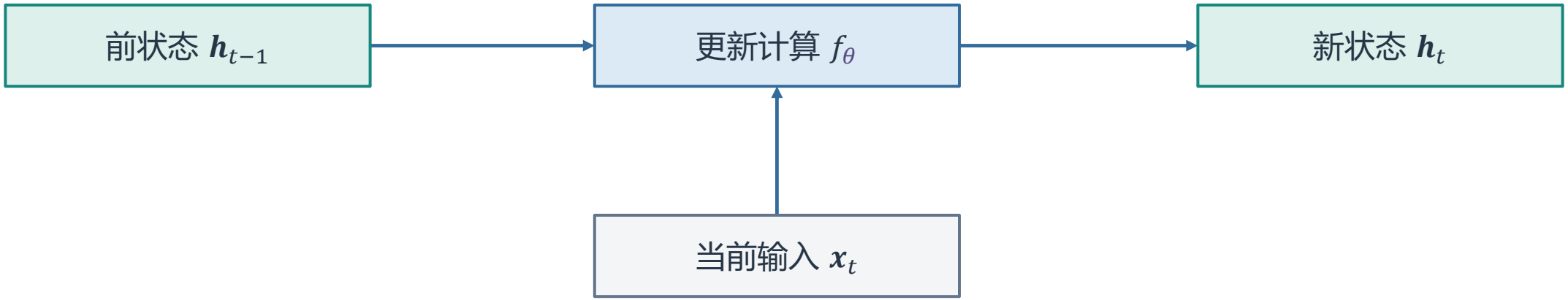
预测位置后移时，窗口随之移动；窗口宽度k保持固定

2 递推状态与参数共享

维护历史表示，在时间位置复用更新规则

隐藏状态与递推更新

- 希望较早信息仍有机会影响当前预测，同时保持每一步历史表示的维度固定
- 为此维护一个内部表示：每读入新元素，就结合此前表示更新一次
- **隐藏状态** $\boldsymbol{h}_t$ 是逐步更新的固定维历史表示
- 当前输入 $\boldsymbol{x}_t$ 与前状态 $\boldsymbol{h}_{t-1}$ 共同形成新状态：$\boldsymbol{h}_t = f_\theta(\boldsymbol{x}_t, \boldsymbol{h}_{t-1})$
- 早期信息可沿状态链间接影响当前表示；状态由任务学习，不保证无损保存全部历史



前后状态均为m维

状态与共享参数

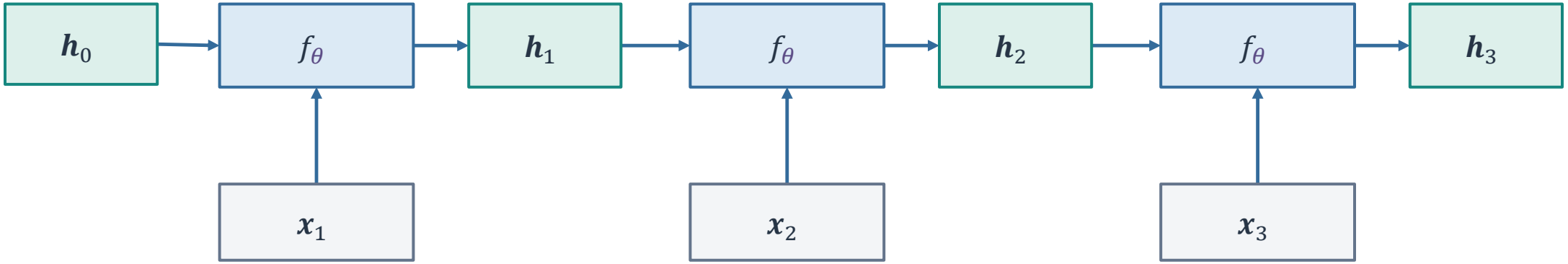
每读入新元素，状态随之更新，各步共用同一更新规则。下面区分状态、参数与展开步数。

对象	随时间发生什么	承担的责任
状态 $\boldsymbol{h}_t$	随当前输入和历史改变	当前序列的运行时表示
参数 θ	同一次前向的各时间位置共用	决定状态怎样更新
展开步数 T	随序列长度增加	增加计算和中间状态

- 同一更新规则可处理不同位置，参数数量不随 T 增长
- 规则相同允许状态不同；可处理更长序列不保证学会长期依赖

RNN的时间展开

- **循环神经网络（RNN）**沿时间重复使用同一个状态更新计算（Cell）
- 每个Cell接收本步输入和前状态，输出新状态；本讲从 $\boldsymbol{h}_0 = \mathbf{0}$ 开始
- 图中各处 f_θ 使用同一组参数，$\boldsymbol{h}_1, \boldsymbol{h}_2, \boldsymbol{h}_3$ 是不同状态



青绿：状态；紫色θ：各步共用；蓝线：前向依赖

3 RNN计算、监督与生成

将抽象更新落实为计算、预测和监督

文本怎样成为RNN的输入？

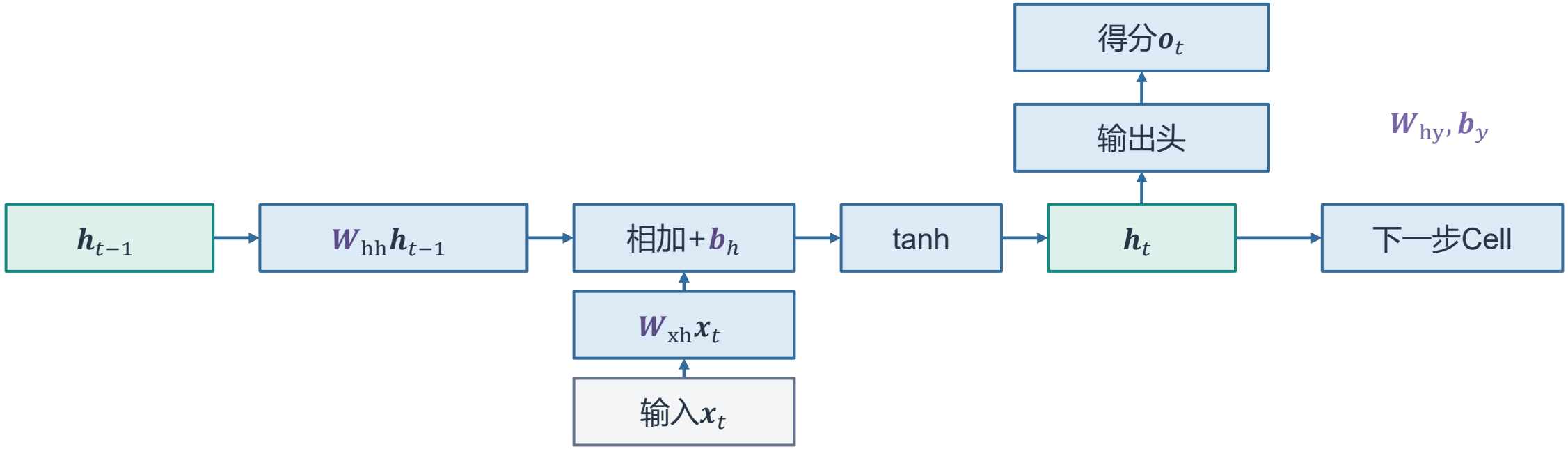
- P10的 x_t 必须是数值向量。文本先切成**词元（token）**，本例采用字符。
- 词表把词元编号。本例a、b的ID为0、1；i_t 表示第t个词元ID，数字只表身份。
- 用可学习的**嵌入（embedding）**表 $\boldsymbol{E} \in \mathbb{R}^{V \times d}$ 把ID查成列向量 $\boldsymbol{x}_t$。V为词表大小，d为输入维数；同词元取同一行。



ID只作身份索引；同一词元查询同一行。输入向量供P10的Cell读取。

RNN单步更新：输入、状态与预测

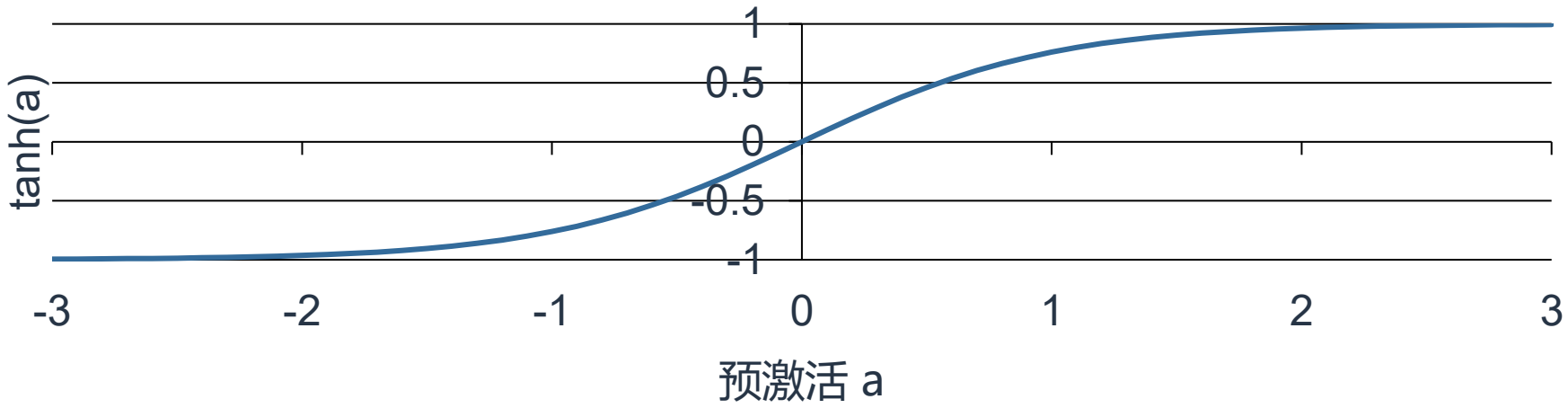
- 当前输入与前状态先变换到同一维度，再相加：$\boldsymbol{a}_t = \boldsymbol{W}_{\mathrm{xh}}\boldsymbol{x}_t + \boldsymbol{W}_{\mathrm{hh}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_h$。
- 只用仿射运算，递推仍是输入向量的仿射组合；因此加入非线性。本讲选 $\boldsymbol{h}_t = \tanh(\boldsymbol{a}_t)$，下一页解释其性质。
- 新状态向后传递，同时送入输出头预测下一词元；各步复用同一组参数。



沿用P10：状态横向传递，输入从下方进入，预测从上方分出。

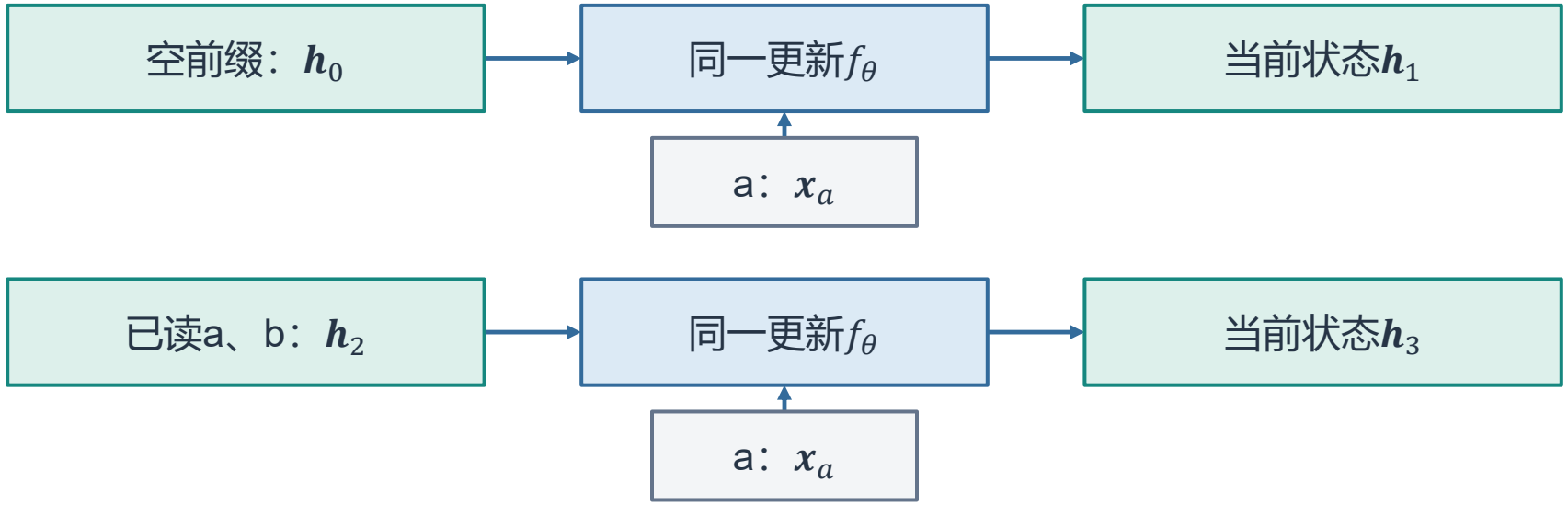
为什么加入非线性，为什么选tanh?

- 给定步数与初态，若更新只含仿射运算，展开后仍是输入向量的仿射组合；非线性使更新能够表达更丰富的关系。
- 本讲采用逐分量作用的tanh，输出在 $(-1,1)$ 内，且 $\tanh(0) = 0$。它可表示正负值，平滑可微，可用于反向传播。
- tanh是基本RNN的一种经典选择，并非任务唯一指定；输入幅值大时曲线趋平，不能据此保证长期记忆或训练稳定。



同一输入为何可能得到不同状态?

- 第1步和第3步都读入a，所以输入向量相同；但此前分别是空前缀与a、b，前状态的来源不同。
- 同一规则使用输入和前状态，故相同输入在不同历史下**可能**形成不同状态。
- 不同历史不保证一定产生不同状态。下一页核对输入、状态和输出的维度。



输入相同不代表状态相同；不同历史也不保证一定产生不同状态。

从输入到预测：维度怎样对应?

对象 / 映射	维度 / 形状	哪个轴发生变化
输入 → 状态	d维 → m维；$\boldsymbol{W}_{\mathrm{xh}}: m \times d$	输入特征映射到状态空间
前状态 → 当前状态	m维 → m维；$\boldsymbol{W}_{\mathrm{hh}}: m \times m$	保持状态维数，更新表示
状态 → 类别得分	m维 → V维；$\boldsymbol{W}_{\mathrm{hy}}: V \times m$	每个词元对应一个得分
两处偏置	$\boldsymbol{b}_h: m$维，$\boldsymbol{b}_y: V$维	与各自输出相加

- 单样本采用列向量；d、m、V分别表示输入维数、状态维数、词表大小。
- 展开时间增加计算和中间状态，不增加独立参数；矩阵形状由d、m、V决定。

状态怎样产生下一词元预测？

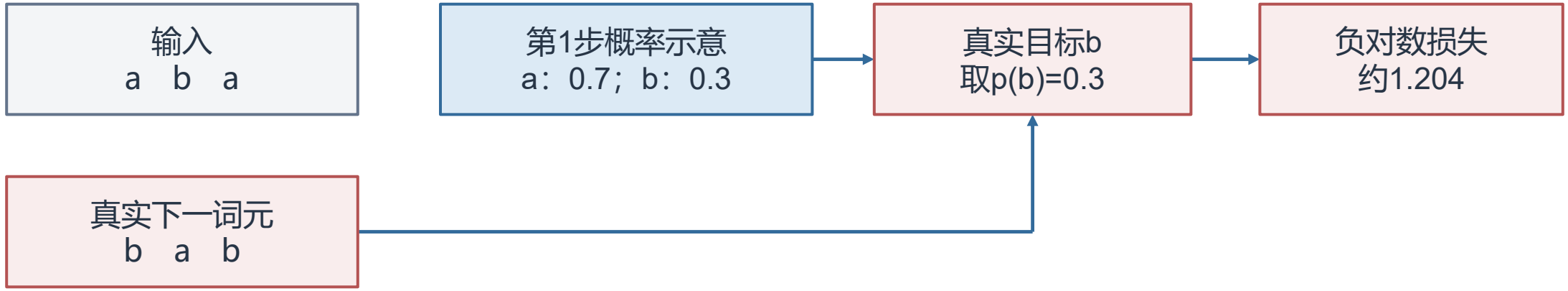
- RNN用状态 $\boldsymbol{h}_t$ 表示已读前缀的影响；输出头给出V类得分 $\boldsymbol{o}_t = \boldsymbol{W}_{\mathrm{hy}}\boldsymbol{h}_t + \boldsymbol{b}_y$。
- 同一位置沿词表类别轴作softmax：$p_{t,j} = \dfrac{\exp(o_{t,j})}{\sum_{k=0}^{V-1}\exp(o_{t,k})}$；这些概率之和为1。
- 得到 $p_{t,j} = P_\theta(i_{t+1} = j \mid i_{1:t})$，对应P5任务。读入第t个词元，预测第t+1个。



示意：已读入第t个词元，输出分布对应第t+1个；状态仍沿时间主路继续传递。

下一词元标签怎样监督预测?

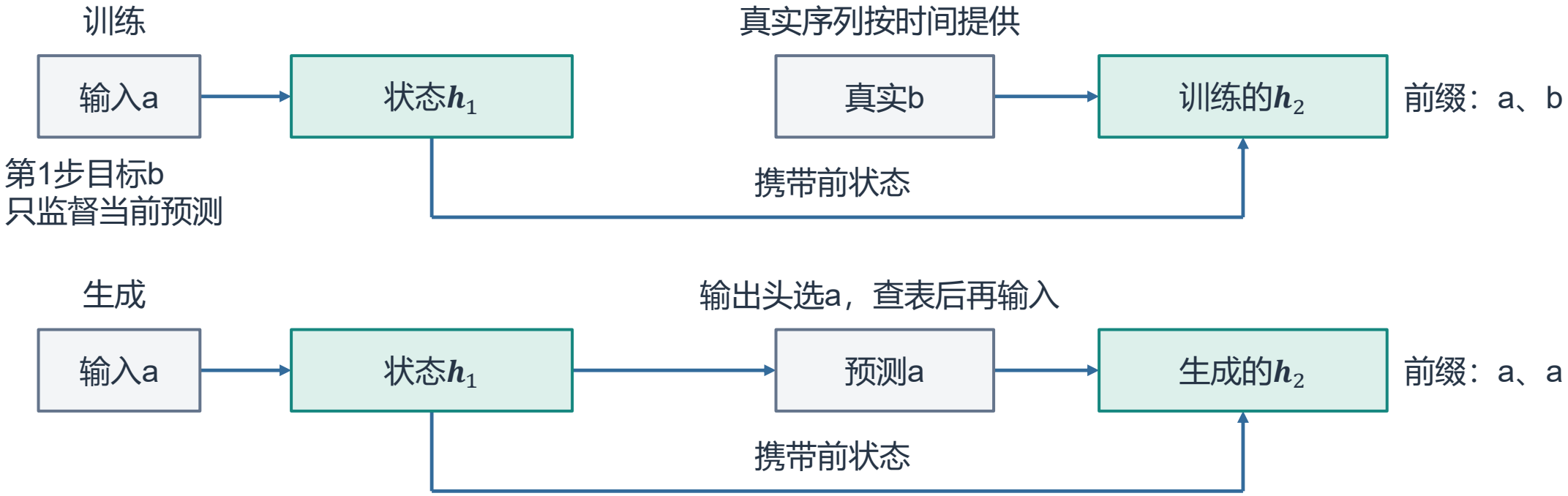
- 真实序列abab对应输入a、b、a和目标b、a、b，即 $y_t = i_{t+1}$；标签向后错一位。
- 单步损失取真实目标的概率：$\ell_t = -\ln p_{t,y_t}$。图中沿用上一页0.7 / 0.3概率示意。
- 本页 S 为序列内有效预测位置集合，$|S| > 0$；学习目标为：
  - $\theta^\star = argmin_\theta L(\theta), L(\theta) = -\frac{1}{|S|}\sum_{t\in S} \ln p_{t,y_t}$
- 交叉熵直接接收logits；训练损失前不做argmax。



只取真实目标的概率计算损失；对全部有效位置取平均，训练损失前不做argmax。

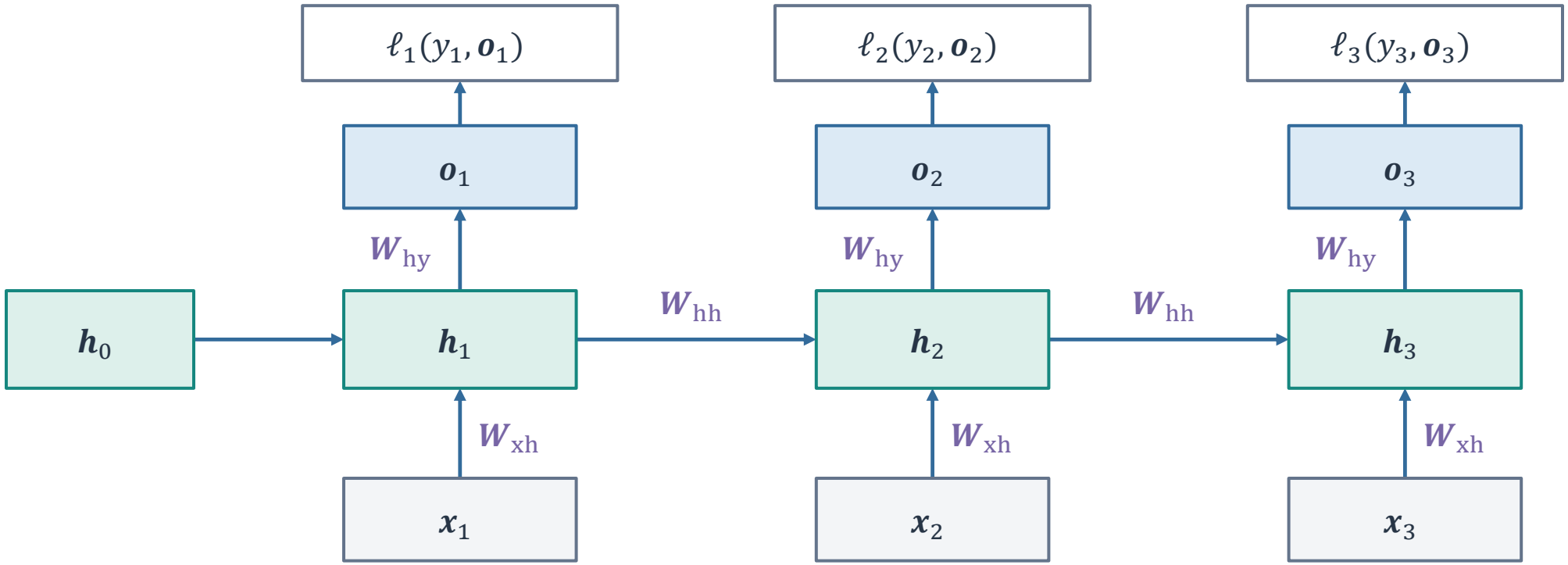
训练输入与生成反馈

- 训练读取真实历史，以下一词元目标监督当前预测；目标到下一步才成为输入
- 生成也从前缀a开始，选出的预测词元反馈为下一步输入；本例用贪心选择



RNN前向计算与损失

- 训练沿真实输入逐步计算损失，再通过反向传播调整共享参数；argmax不放在训练损失之前
- 现在已能评价预测，下一步要解释：后续误差怎样影响此前的更新计算？



各步得分对照真实下一词元计算损失；状态沿时间向右更新

4 时间反向传播与共享参数梯度

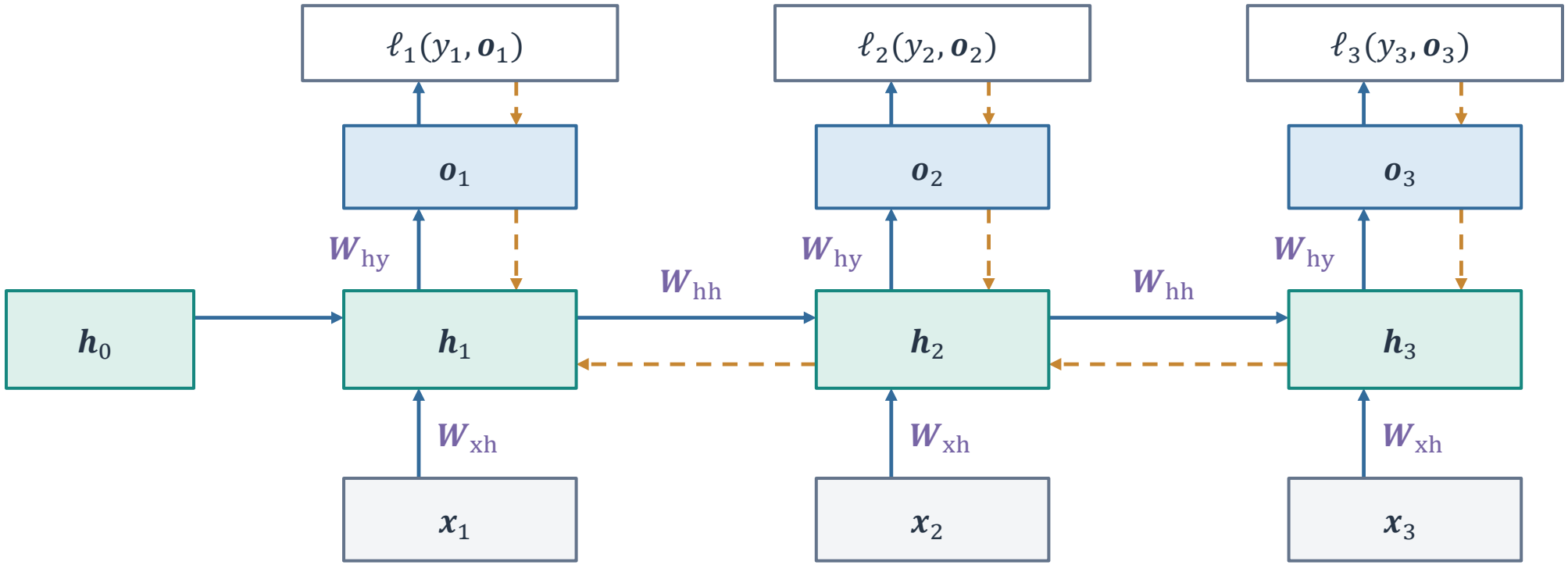
追溯后续误差，学习同一更新规则

跨时间的学习问题

- 若第3步预测错误，输出头的参数需要调整；但错误也可能与形成 $\boldsymbol{h}_3$ 的历史表示有关
- 因为 $\boldsymbol{h}_3$ 依赖 $\boldsymbol{h}_2$，而 $\boldsymbol{h}_2$ 又依赖 $\boldsymbol{h}_1$，末步损失能够影响较早计算
- 需要同时回答：影响沿哪条时间路径返回？同一参数多次使用的贡献如何合并？
- 前向输入仍限于可见历史；反向误差信号用于学习更新规则

通过时间反向传播（BPTT）

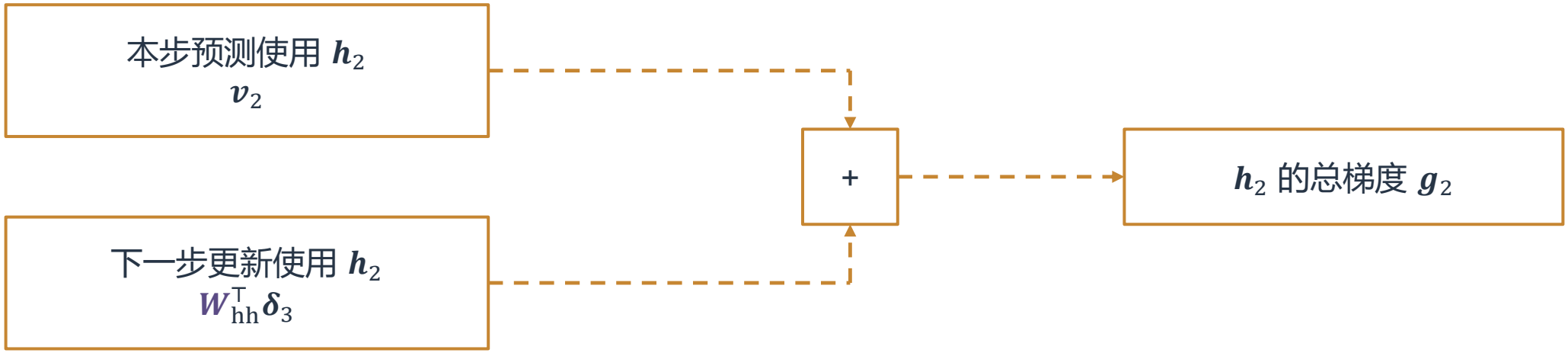
- **通过时间反向传播（BPTT）**沿展开图应用链式法则；框内为状态值，后续损失的梯度可回传至较早状态
- 例：从 ℓ_3 经 $\boldsymbol{o}_3, \boldsymbol{h}_3, \boldsymbol{h}_2$ 回到 $\boldsymbol{h}_1$；模型前向仍从左向右



实线：前向依赖；虚线：梯度回传

状态梯度的两路汇合

- 上页的 $\boldsymbol{h}_2$ 同时用于本步预测和下一步状态更新



同一状态的两路使用，产生的梯度相加

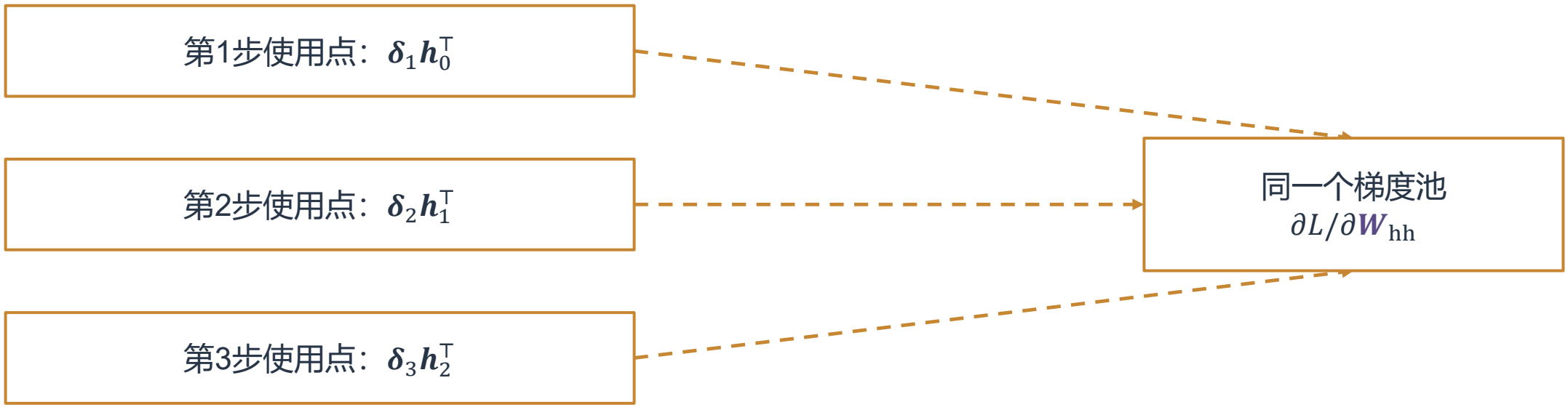
- 记状态总梯度 $\boldsymbol{g}_t = \partial L/\partial \boldsymbol{h}_t$，预激活总梯度 $\boldsymbol{\delta}_t = \partial L/\partial \boldsymbol{a}_t$；本步分支为 $\boldsymbol{v}_t = \boldsymbol{W}_{\mathrm{hy}}^{\mathrm{T}}(\partial L/\partial \boldsymbol{o}_t)$
- 两路汇合：$\boldsymbol{g}_t = \boldsymbol{v}_t + \boldsymbol{W}_{\mathrm{hh}}^{\mathrm{T}}\boldsymbol{\delta}_{t+1}$
- 末步无后续状态，$\boldsymbol{g}_T = \boldsymbol{v}_T$；没有本步损失时，$\boldsymbol{v}_t = \boldsymbol{0}$

tanh激活的梯度回传

- **局部导数**
  - 前向 $\boldsymbol{h}_t = \tanh(\boldsymbol{a}_t)$ 逐分量作用，标量导数为 $d\tanh(a)/da = 1 - \tanh^2(a)$
  - 预激活总梯度为 $\boldsymbol{\delta}_t = \boldsymbol{g}_t \odot (\mathbf{1} - \boldsymbol{h}_t \odot \boldsymbol{h}_t)$
    - $\odot$ 表示逐分量相乘，$\mathbf{1}$ 是 m 维全1向量；状态与两种梯度均为 m 维
    - 某分量 $h = 0.8, g = 0.5$ 时，$\delta = 0.5(1 - 0.8^2) = 0.18$
- **回传次序**
  - 末步先求 $\boldsymbol{g}_T = \boldsymbol{v}_T$，再求 $\boldsymbol{\delta}_T$；随后用它求 $\boldsymbol{g}_{T-1}$，再求 $\boldsymbol{\delta}_{T-1}$，依次回传

共享参数的梯度累积

- 已知各步误差信号，现求它们对同一共享权重的贡献；各处贡献相加后统一更新
- 循环项 $\boldsymbol{W}_{\mathrm{hh}}\boldsymbol{h}_{t-1}$ 对应第 t 处贡献 $\boldsymbol{\delta}_t\boldsymbol{h}_{t-1}^{\top}$，其中 $\boldsymbol{\delta}_t$ 已含本步与后续影响
- 因而 $\dfrac{\partial L}{\partial \boldsymbol{W}_{\mathrm{hh}}} = \sum_{t=1}^{T} \boldsymbol{\delta}_t\, \boldsymbol{h}_{t-1}^{\top}$



各项为 $m\times m$ 外积；汇总后更新同一组参数

琥珀虚线：各使用点贡献汇入同一梯度池

线性递推的前向计算

为看清时间回传与共享求和，暂用标量和恒等激活，保留递推、误差回传与贡献累积。

t	输入 x_t	状态 h_t
1	1	1
2	2	2.5
3	0	1.25

- 另取标量线性模型 $h_t = wh_{t-1} + ux_t + b$，设 $h_0 = 0, w = 0.5, u = 1, b = 0$；$w, u$ 分别对应循环、输入权重
- 仅末步目标为0：$L = \frac{1}{2}h_3^2 = 0.78125$。恒等激活下 $\delta_t = g_t$，故 $\partial L/\partial w = \sum_t g_t\, h_{t-1}$

共享权重梯度的数值计算

使用位置 t	状态梯度 g_t	前状态 h_{t-1}	对同一 w 的贡献 $g_t h_{t-1}$
1	0.3125	0	0
2	0.625	1	0.625
3	1.25	2.5	3.125

- 仅末步有损失：$g_3 = h_3 = 1.25$；此前 $g_t = w g_{t+1}$，故 $g_2 = 0.5 \times 1.25 = 0.625$，$g_1 = 0.5 \times 0.625 = 0.3125$
- 累积后统一更新同一个 w：$\partial L / \partial w = 0 + 0.625 + 3.125 = 3.75$

5 长期依赖与基本RNN的局限

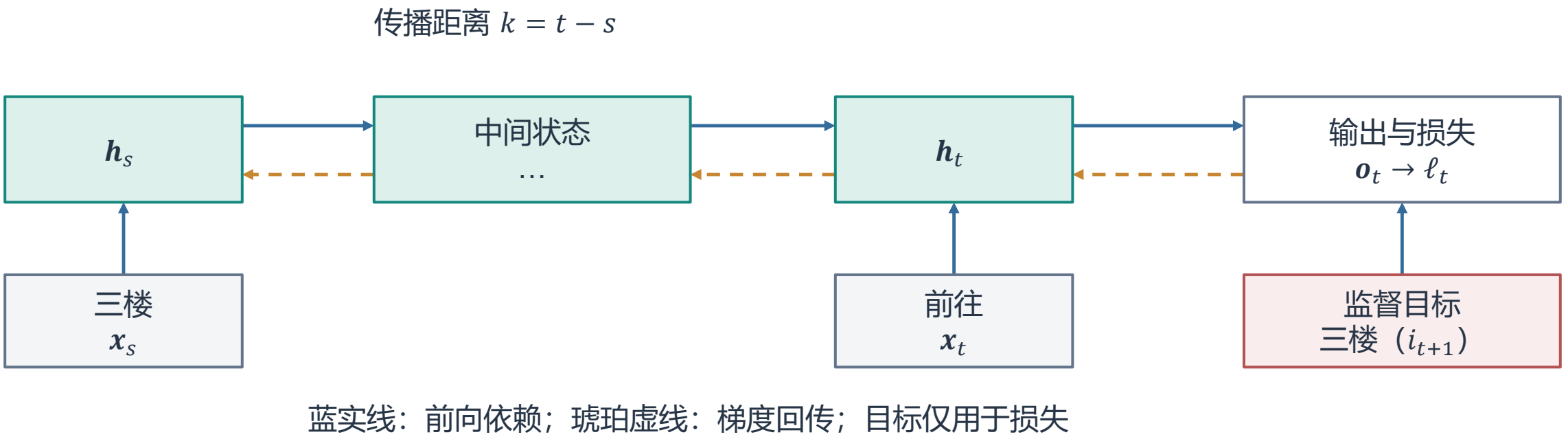
分别分析信息保留与梯度传播

长期依赖问题

- 按词示意切分：“通知：会议改在**三楼**会议室举行。……大家处理完手头工作，按通知**前往**【待预测】会议室。”
- 当前预测需要利用相隔较多时间步的历史信息，这种依赖称为**长期依赖**
- 已读“三楼”的位置为 s，“前往”的位置为 t；真实下一词元 i_{t+1} 为“三楼”
- 中间未重复地点，较早通知需经 $\boldsymbol{h}_s, \ldots, \boldsymbol{h}_t$ 影响预测；递推通路不保证信息始终可用或容易学到

历史信息保留与梯度传播

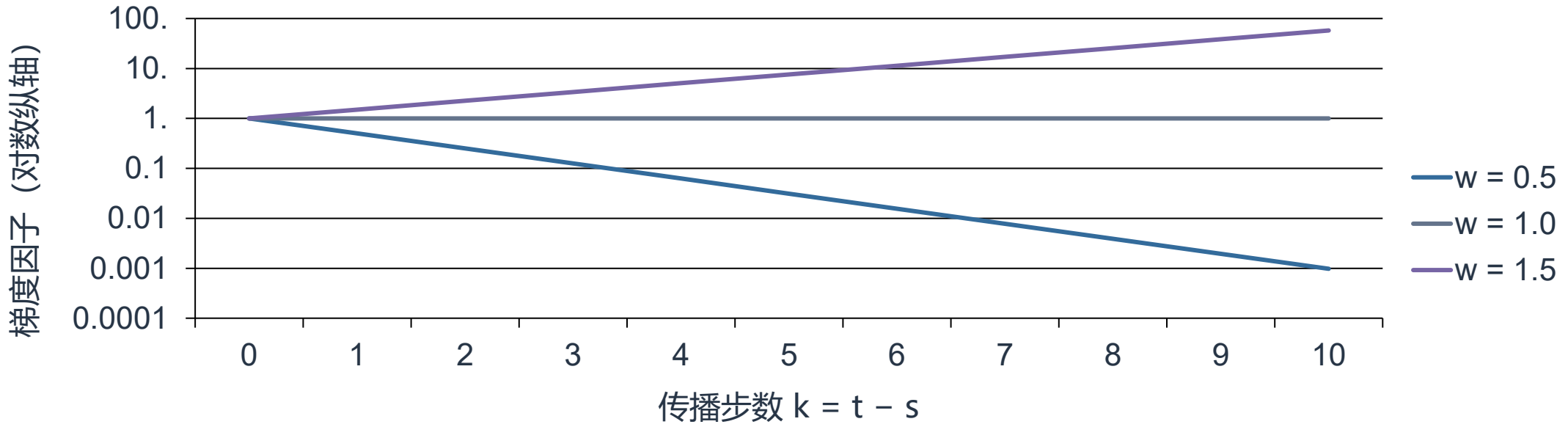
- **前向信息保留**：从 $\boldsymbol{h}_s$ 到 $\boldsymbol{h}_t$ 的更新，可能丢失早期线索或使其难以读出
- **反向学习信号**：末端预测损失 ℓ_t 沿同一依赖链影响较早计算；局部导数连乘可能削弱或放大学习信号



蓝实线：前向依赖；琥珀虚线：梯度回传；目标仅用于损失

梯度消失与爆炸

- 沿上页 $h_s \to h_t$，传播距离 $k = t - s$；线性标量例中 $\partial h_t / \partial h_s = w^k$
- 一般tanh RNN还包含各步激活导数，实际尺度取决于参数、状态和路径
- 裁剪限制过大梯度的范数，不能据此认为长期记忆或梯度消失已解决



基本RNN的局限与改进需求

- 前向从 $\boldsymbol{h}_s$ 到 $\boldsymbol{h}_t$：更新可能弱化早期线索；可递推不等于可靠长期记忆
- 反向从 ℓ_t 回到较早计算：梯度可能消失或爆炸；裁剪限制过大梯度，不能恢复已丢失的信息
- 需要更有选择地保留、更新与遗忘状态，门控结构由下一讲LSTM与GRU展开
- 本讲另一个实际问题是长序列的计算成本：分段和截断控制回传范围，本身不保证解决长期依赖

6 长序列分段与状态管理

长序列指总时间步较多；长期依赖指预测需利用相隔较多时间步的历史信息，两者均无统一步数门槛。分段与截断控制计算及回传范围，不保证解决长期依赖。

长序列的分段处理

将a、b、a、b分成连续片段，输入仍与原序列的下一词元目标对齐。

连续片段	输入	每步目标
片段1	a，b	b，a
片段2	a	b

- 片段1末状态可作为片段2初始状态，后者最后预测第4个词元b

状态携带与重置

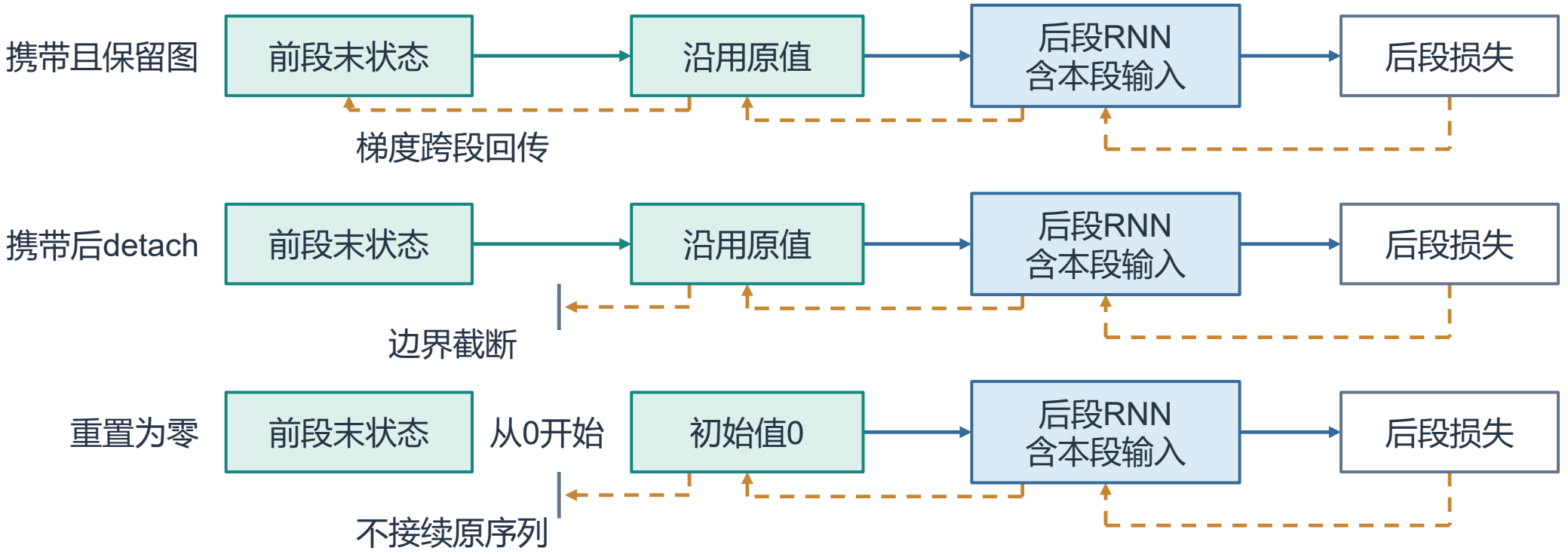
是否携带末状态，取决于片段是否连续且属于同一条序列。

数据关系	状态处理	原因
不相关的独立序列	各自初始化状态	不引入其他样本的历史
同一流按顺序切成连续片段	可以携带末状态	后段需要前段的历史
片段被打乱或批量行换了流	重置或重新匹配状态	原状态已不对应当前前缀

- 本讲采用零初始化；不能只因片段出现在相邻批次就携带状态

状态值与梯度连接

- **detach**保留状态值，切断边界处的跨段梯度；后段仍受历史状态数值影响
- 后段经RNN计算才产生损失；携带要求连续同流、批量行身份匹配



青绿：状态值；蓝：计算依赖；琥珀虚线：梯度；灰线：截断

截断BPTT的训练流程

1. 为一段连续流初始化状态，读取当前片段及对齐标签
2. 以当前状态前向，计算本片段有效位置损失
3. 在该片段允许的图内反向传播，再更新共享参数
4. 分离并携带末状态，读取同流下一片段及标签，返回第2步
5. 遇到新独立序列或不连续跳转时，按约定重置状态

- 分段限制回传长度与存储成本，得到的是截断梯度，不是整段序列的完整BPTT

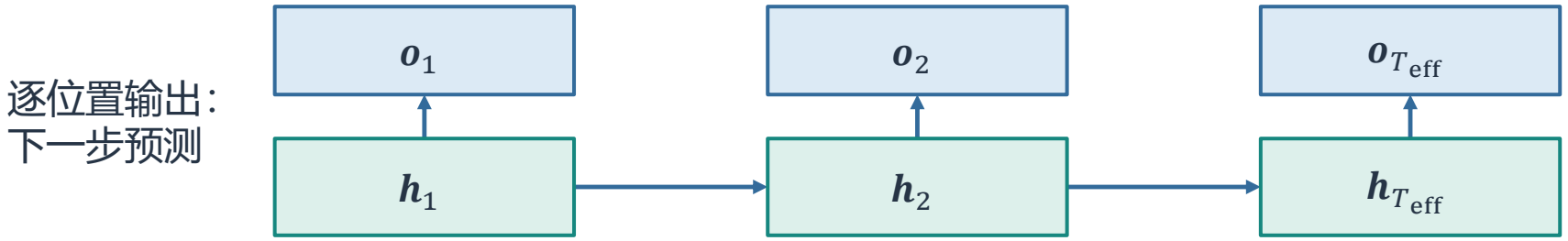
选讲：批量维度与有效位置

对象	本讲存储形状	轴或位置的含义
词元ID	$B \times T$	B 条序列，各有 T 个存储位置
嵌入输入	$B \times T \times d$	每个位置有 d 维向量
状态序列	$B \times T \times m$	每个位置有 m 维状态

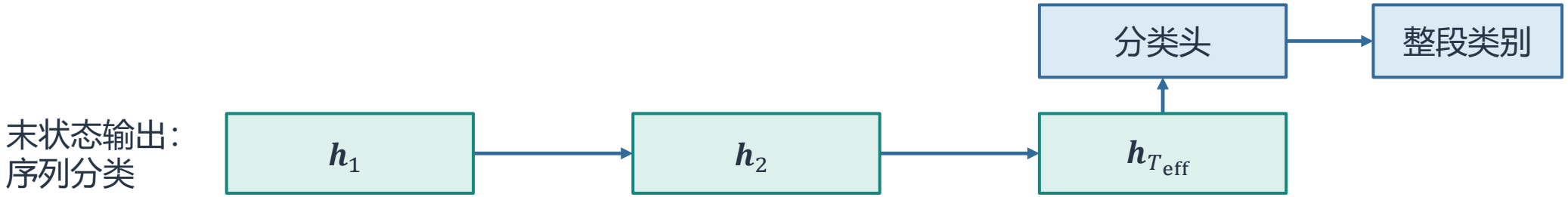
- 各行独立递推；跨片段携带须保持同流身份。单样本仍用列向量，表中仅规定批量存储轴
- 仅对有效位置集合 S（序列编号 n、时间 t）求平均：$L = -|S|^{-1} \sum_{(n,t)\in S} \ln p_{n,t,y_{n,t}}$，$|S| > 0$；填充不计入目标或分母

选讲：序列预测与分类的状态使用

- 同一状态序列可用于不同任务：下一步预测逐位置输出；序列分类用真实末有效状态接分类头
- 输出头、损失与评价由任务决定；末状态保留历史的能力受模型与训练限制



图示末有效步数为3；各步状态随当前输入与前状态递推



图示末有效步数为3；各步状态随当前输入与前状态递推

连续信号可在任务指定位置接回归头；右侧填充不属于真实末位置

本讲总结

- **为什么需要RNN?** 固定窗口只直接读取有限历史；RNN用当前输入与前状态递推形成历史表示，各时间位置共享更新规则
- **RNN怎样学习?** 由真实下一词元计算损失，BPTT沿时间回传误差，再汇总同一参数各使用位置的梯度贡献
- **为什么还需要LSTM/GRU?** 基本RNN难以稳定保留远期线索，反向梯度也可能消失或爆炸；这些困难提出选择性保留、更新与遗忘信息的需求

训练读取真实历史，生成反馈所选词元；分段时状态值与梯度连接分别管理，截断本身不保证解决长期依赖。

备选：detach的状态值与梯度

第2步之后的处理	第3步初始状态	第3步输出	末步损失对 w 的梯度
携带且保留图	2.5	1.25	3.75
携带后detach	2.5	1.25	3.125
重置为零	0	0	0

- 沿用线性例：输入1、2 | 0，$w = 0.5, u = 1, b = 0$，仅末步损失 $L = h_3^2/2$；对照期间不更新参数
- detach把 h_2 视为固定输入：去掉前段贡献0.625，保留后段贡献 $g_3 h_2 = 3.125$，所以 w 仍可学习；重置的零值只因本例初始状态、x_3, b 与目标均为0

扩展：HMM与RNN的状态

- **隐马尔可夫模型（HMM）**用离散隐状态及其转移、观测生成概率描述序列
- 基本RNN用输入和前状态确定性计算连续状态，再通过输出头构造所需预测
- “隐藏”在两种模型中承担不同含义，不能只凭名称把RNN状态当作HMM的离散状态
- 两者都能组织时间依赖，具体建模假设和学习方法需要分别说明
- 本讲只作扩展定位，HMM的完整推断和参数学习不在当前范围内