Deep Retrieval

王树森

http://wangshusen.github.io/

Deep Retrieval

- 经典的双塔模型把用户、物品表示为向量，线上做最近邻查找。
- Deep Retrieval [1] 把物品表征为路径（path），线上查找用户最匹配的路径。
- Deep Retrieval 类似于阿里的 TDM [2]。

参考文献：

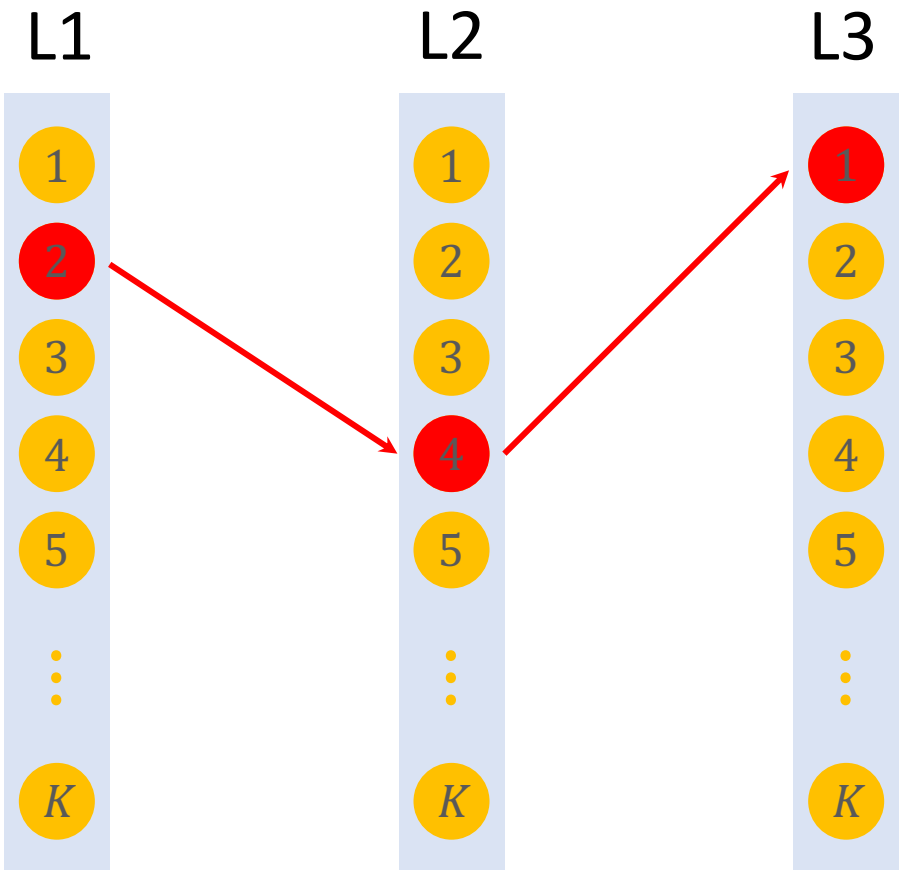
1. Weihao Gao et al. Learning A Retrievable Structure for Large-Scale Recommendations. In *CIKM*, 2021.
2. Han Zhu et al. Learning Tree-based Deep Model for Recommender Systems. In *KDD*, 2018.

Outline

1. 索引：
   - 路径 → List<物品>
   - 物品 → List<路径>
2. 预估模型：神经网络预估用户对路径的兴趣。
3. 线上召回：用户 → 路径 → 物品。
4. 训练：
   - 学习神经网络参数。
   - 学习物品表征（物品 → 路径）。

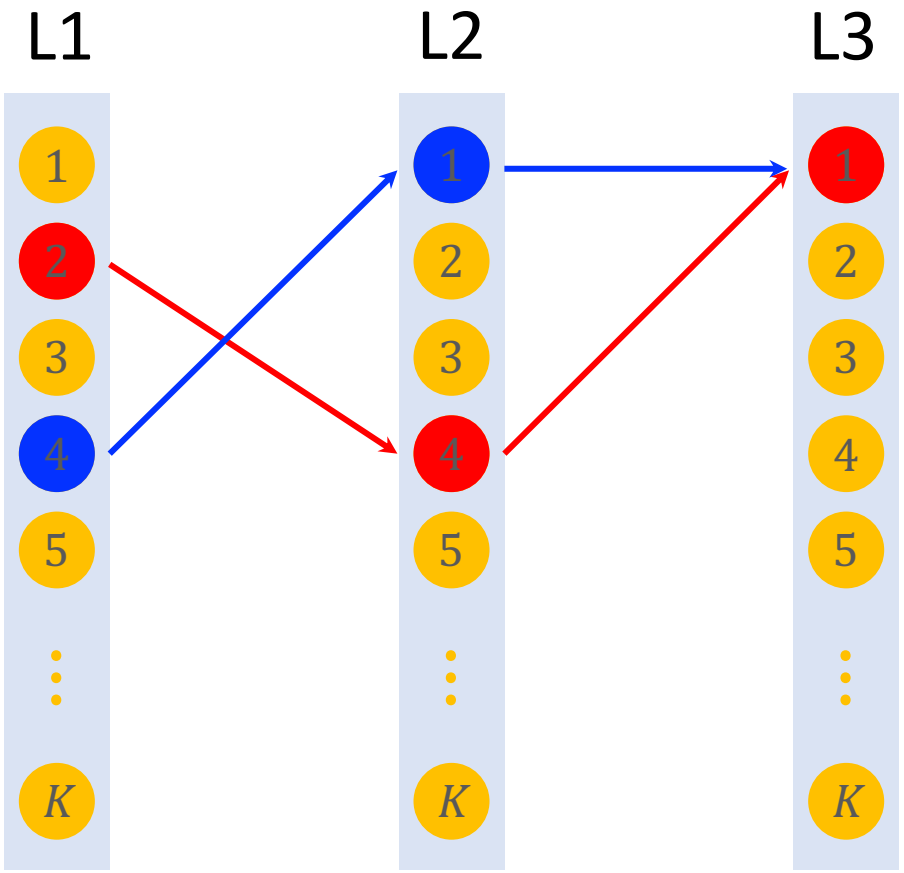
索引

物品表征为路径



- 深度: depth = 3。
- 宽度: width = K。
- 把一个物品表示为一条路径（path），比如 [2, 4, 1]。

物品表征为路径



- 深度: depth = 3。
- 宽度: width = K。
- 把一个物品表示为一条路径（path），比如 [2, 4, 1]。
- 一个物品可以表示为多条路径，比如 {[2, 4, 1], [4, 1, 1]}。

物品到路径的索引

索引：item → List⟨path⟩

- 一个物品对应多条路径。
- 用 3 个节点表示一条路径：path = $[a, b, c]$。

索引：path → List⟨item⟩

- 一条路径对应多个物品。

预估模型

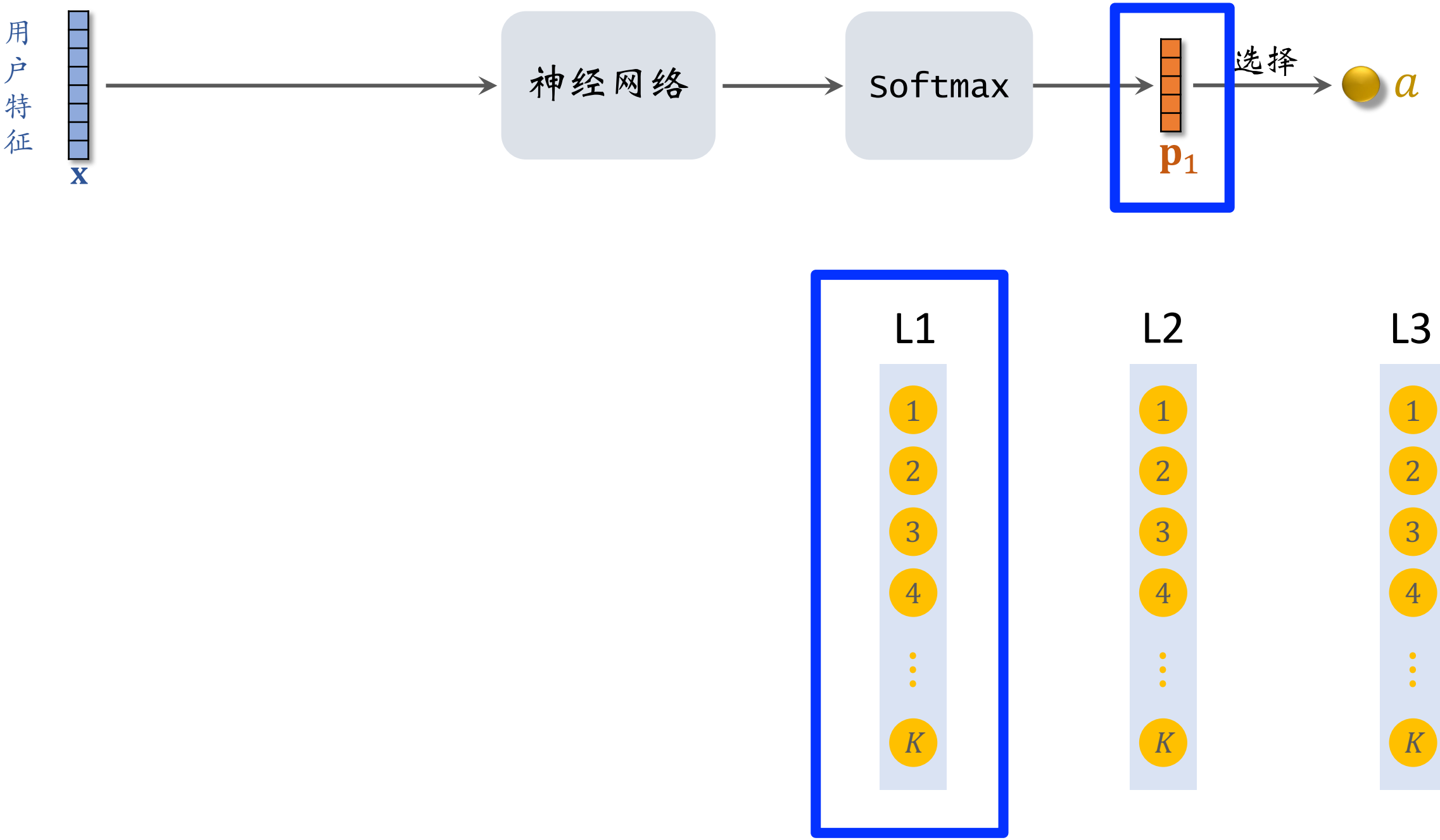
预估用户对路径的兴趣

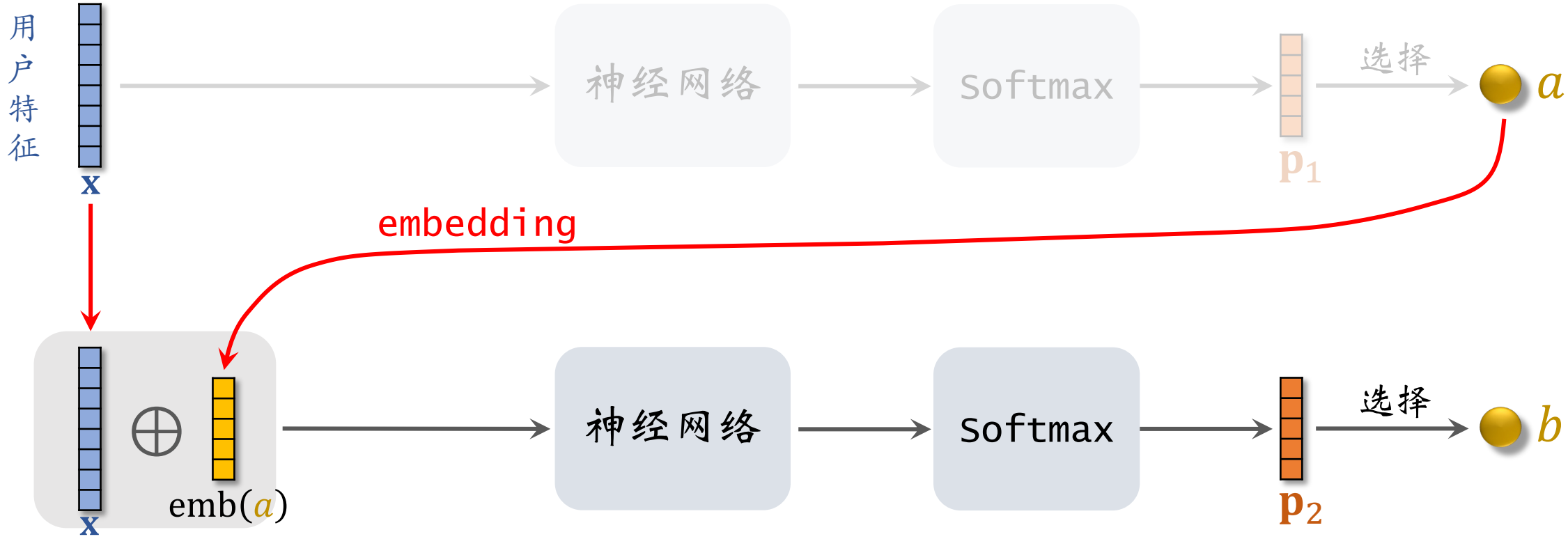
- 用 3 个节点表示一条路径：$\text{path} = [a, b, c]$。
- 给定用户特征 $\mathbf{x}$，预估用户对节点 a 的兴趣 $p_1(a|\mathbf{x})$。
- 给定 $\mathbf{x}$ 和 a，预估用户对节点 b 的兴趣 $p_2(b|a;\mathbf{x})$。
- 给定 $\mathbf{x}, a, b$，预估用户对节点 c 的兴趣 $p_3(c|a,b;\mathbf{x})$。

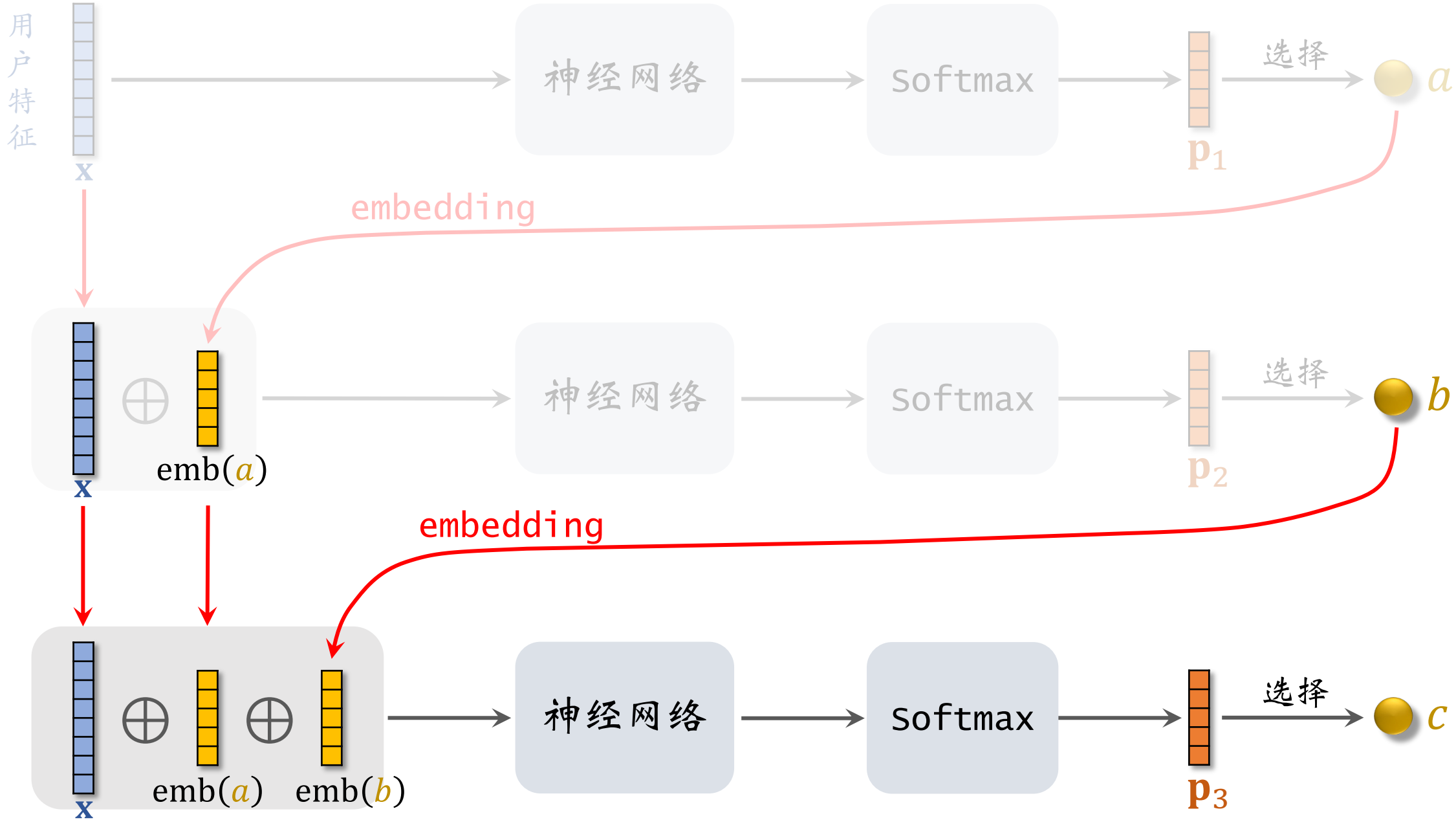
预估用户对路径的兴趣

- 用 3 个节点表示一条路径：$\text{path} = [a, b, c]$。
- 给定用户特征 $\mathbf{x}$，预估用户对节点 a 的兴趣 $p_1(a|\mathbf{x})$。
- 给定 $\mathbf{x}$ 和 a，预估用户对节点 b 的兴趣 $p_2(b|a;\mathbf{x})$。
- 给定 $\mathbf{x}, a, b$，预估用户对节点 c 的兴趣 $p_3(c|a,b;\mathbf{x})$。
- 预估用户对 $\text{path} = [a, b, c]$ 兴趣：

$$p(a,b,c|\mathbf{x}) = p_1(a|\mathbf{x}) \times p_2(b|a;\mathbf{x}) \times p_3(c|a,b;\mathbf{x}).$$


用户特征
x
神经网络
Softmax
p_1
选择
a
L1
L2
L3
1
2
3
4
K


用户特征
x
神经网络
Softmax
p_1
选择
a
embedding
$\oplus$
emb(a)
x
神经网络
Softmax
p_2
选择
b


用户特征
x
神经网络
Softmax
p_1
选择
a
embedding
x
emb(a)
神经网络
Softmax
p_2
选择
b
embedding
x
emb(a)
emb(b)
神经网络
Softmax
p_3
选择
c

线上召回

线上召回

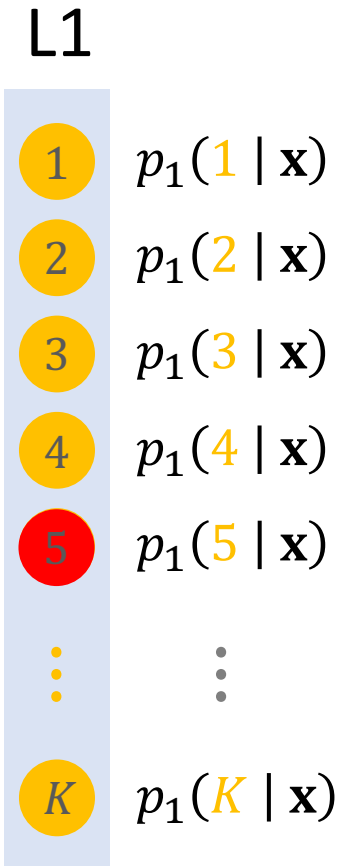
召回：用户 → 路径 → 物品

- 第一步：给定用户特征，用 beam search 召回一批路径。
- 第二步：利用索引“ path → List⟨item⟩”，召回一批物品。
- 第三步：对物品做打分和排序，选出一个子集。

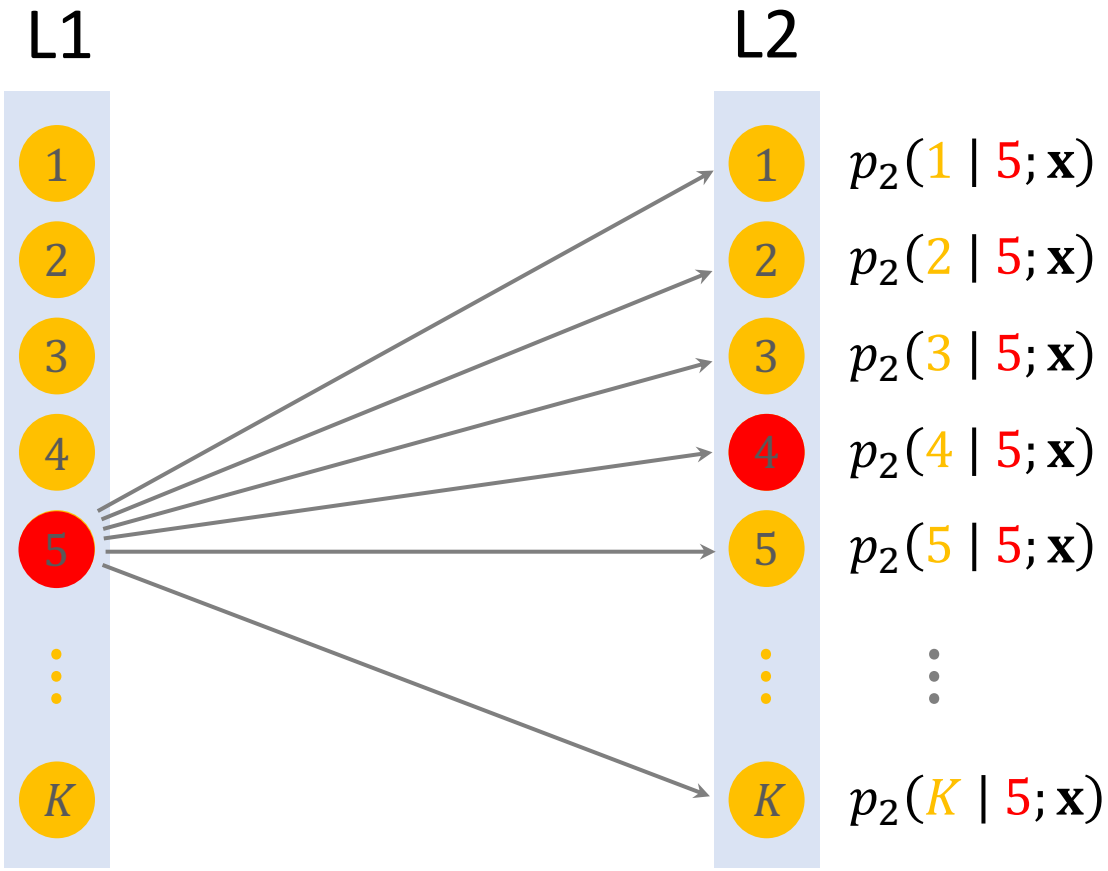
Beam Search

- 假设有 3 层，每层 K 个节点，那么一共有 K^3 条路径。
- 用神经网络给所有 K^3 条路径打分，计算量太大。
- 用 beam search，可以减小计算量。
- 需要设置超参数 beam size。

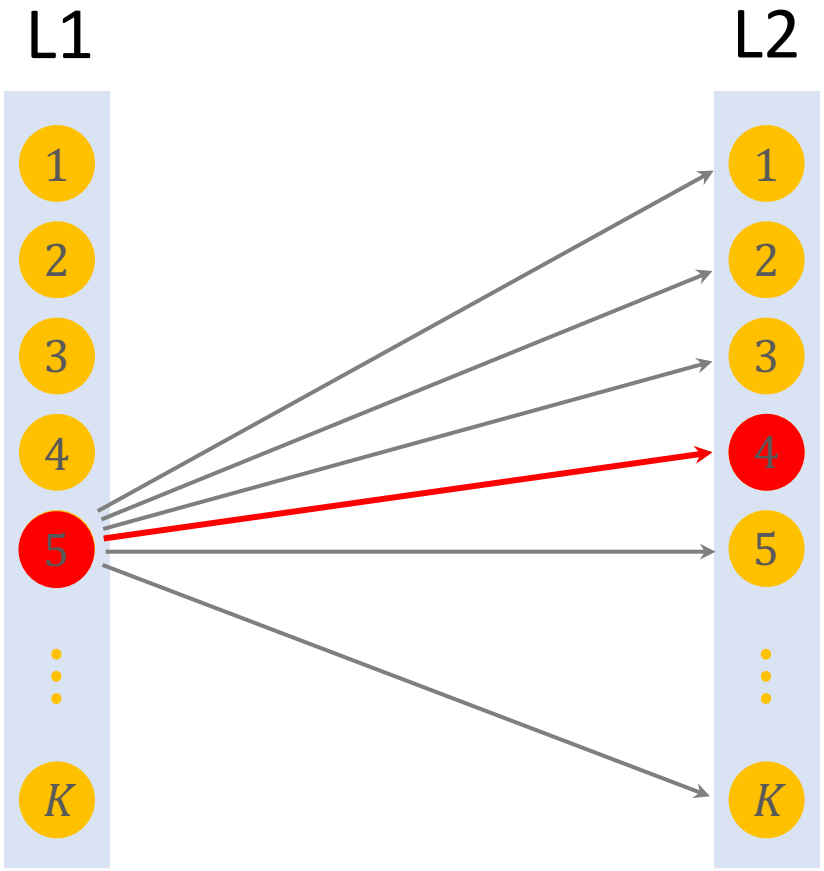
Beam Search (size = 1)



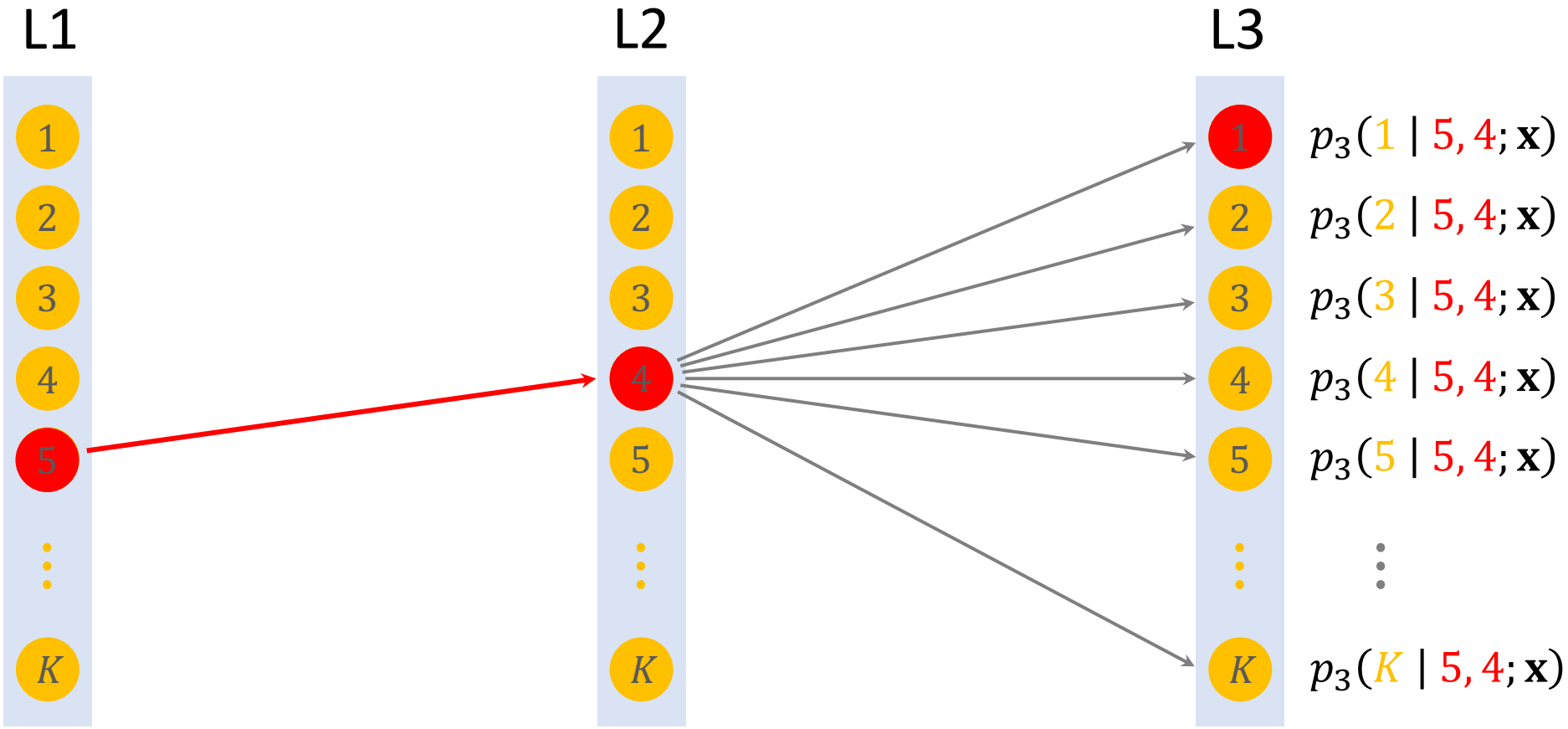
Beam Search (size = 1)



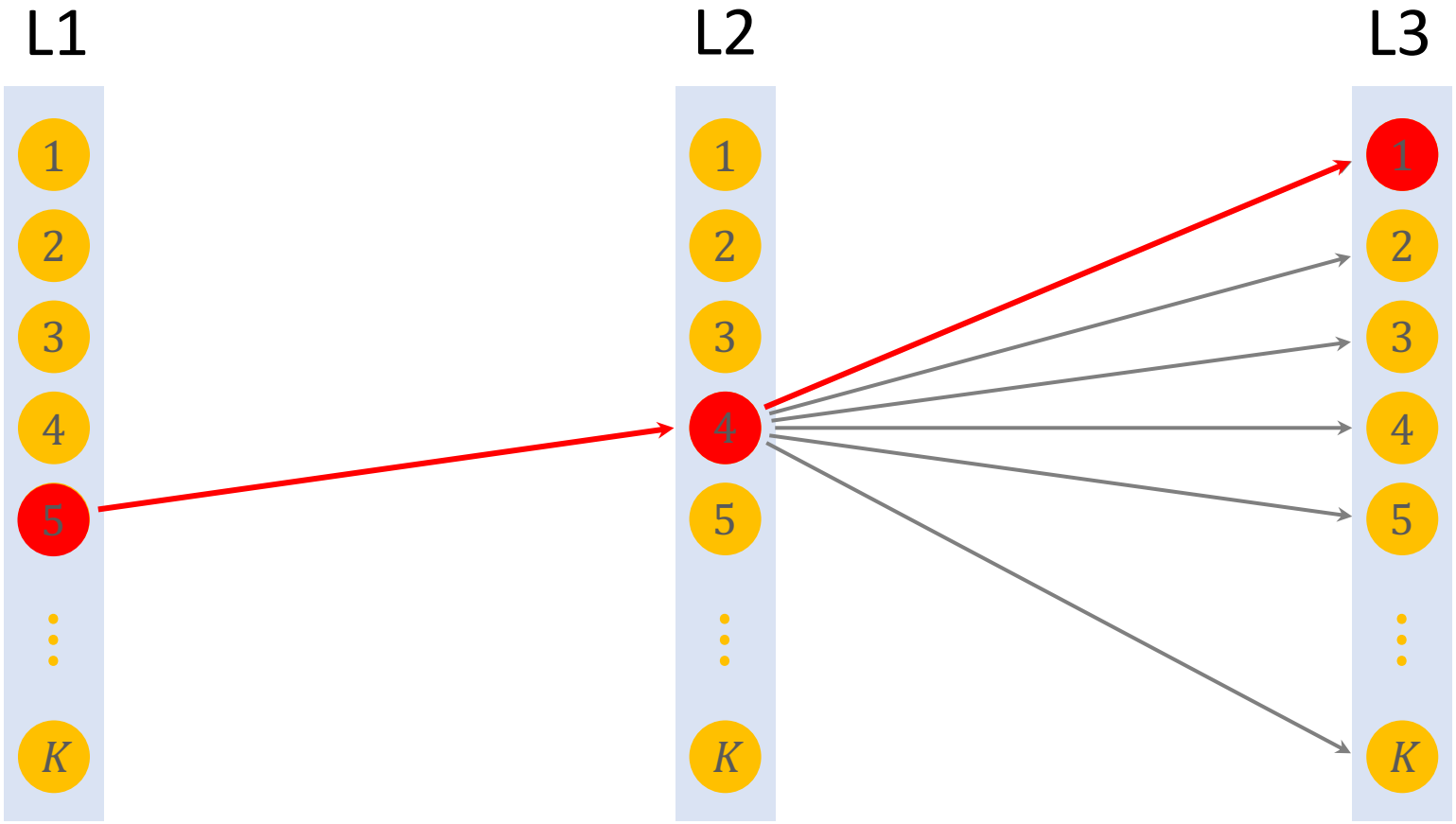
Beam Search (size = 1)



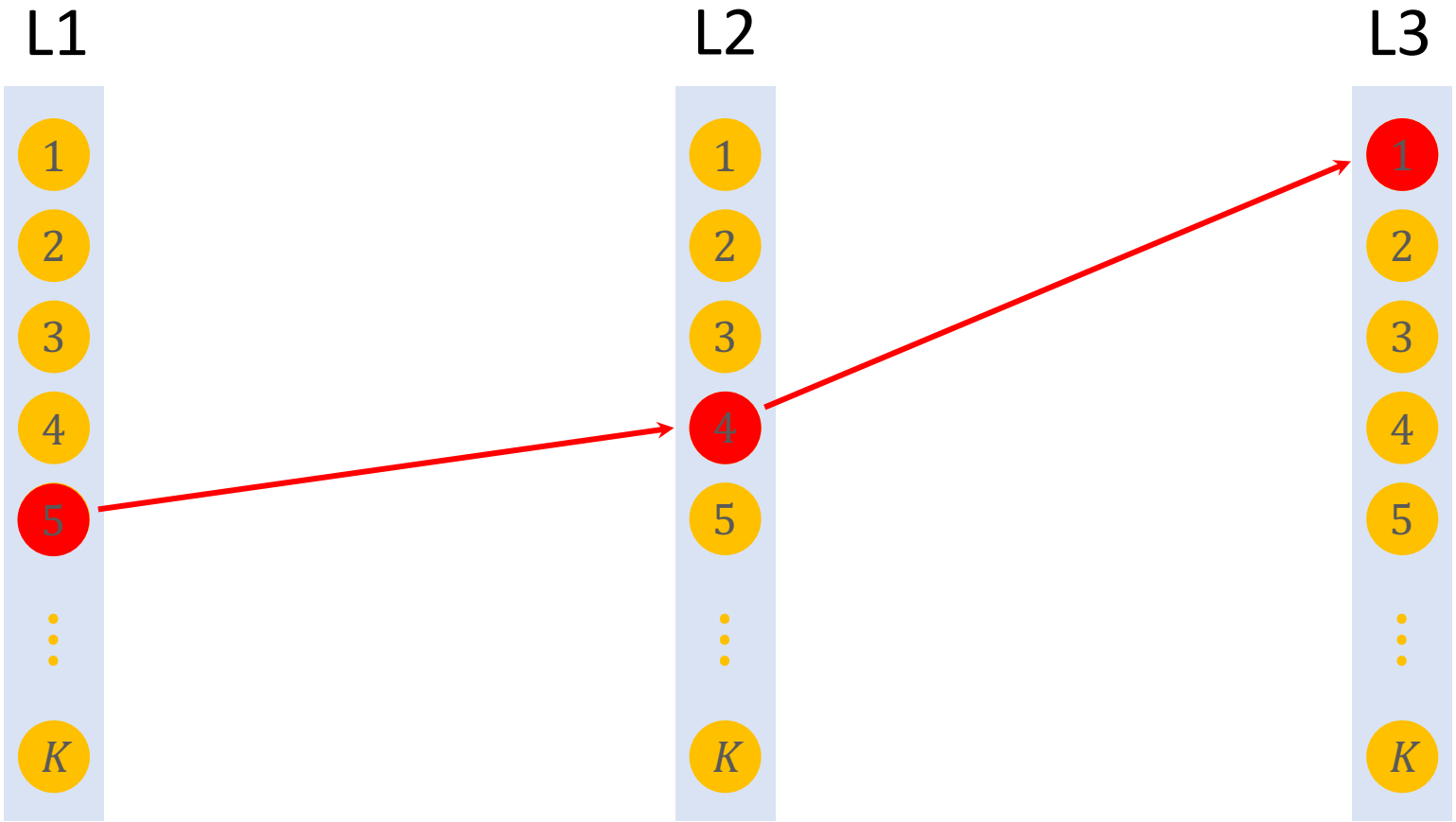
Beam Search (size = 1)



Beam Search (size = 1)



Beam Search (size = 1)



选中路径 path = [5, 4, 1]

Beam Search

- 用户对 $\text{path} = [a, b, c]$ 兴趣：

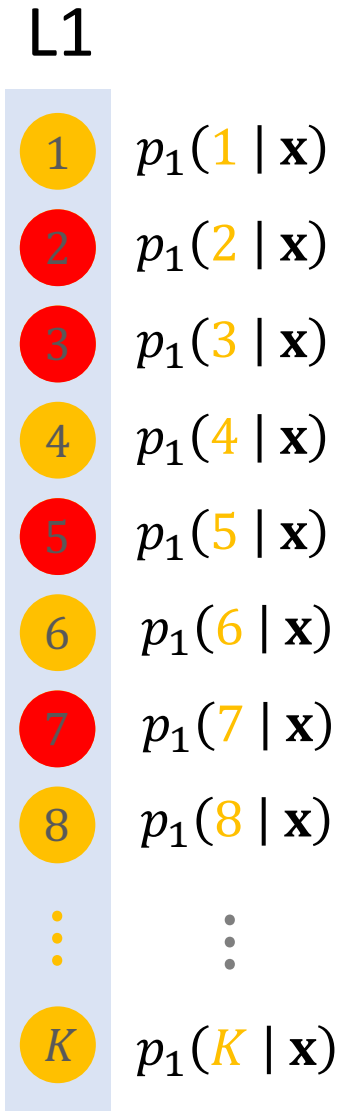
$$p(a, b, c|\mathbf{x}) = p_1(a|\mathbf{x}) \times p_2(b|a; \mathbf{x}) \times p_3(c|a, b; \mathbf{x}).$$

- 最优的路径：

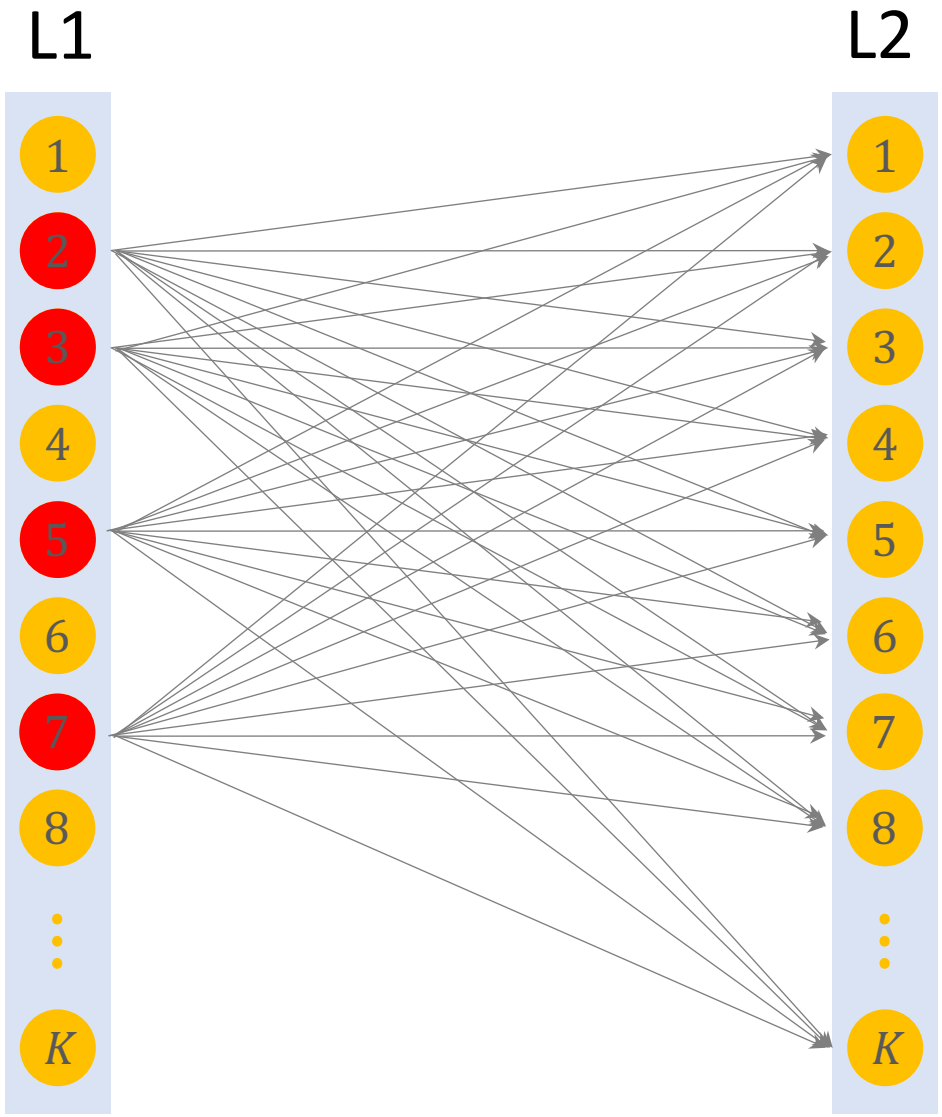
$$[a^\star, b^\star, c^\star] = \underset{a,b,c}{\text{argmax}} \; p(a, b, c \mid \mathbf{x})$$

- 贪心算法（beam size = 1）选中的路径 $[a, b, c]$ 未必是最优的路径。

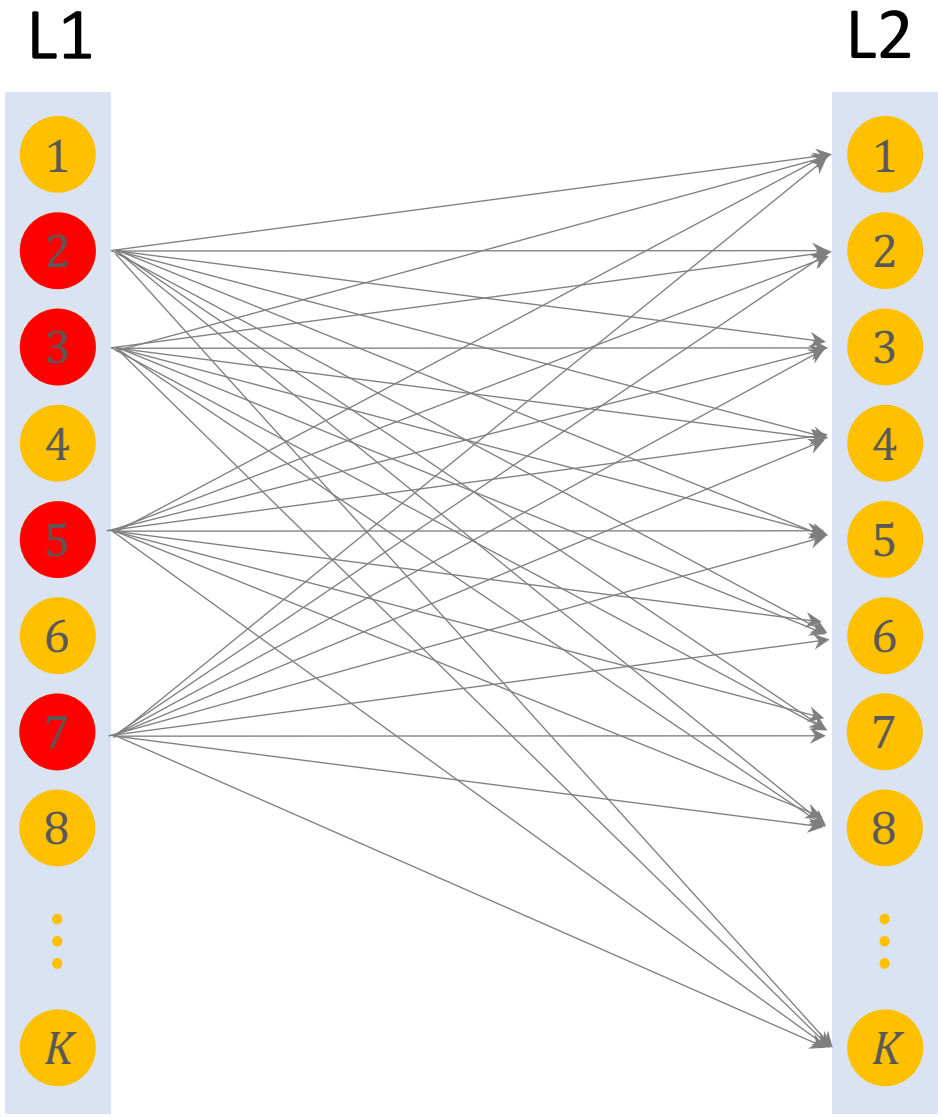
Beam Search (size = 4)



Beam Search (size = 4)



Beam Search (size = 4)

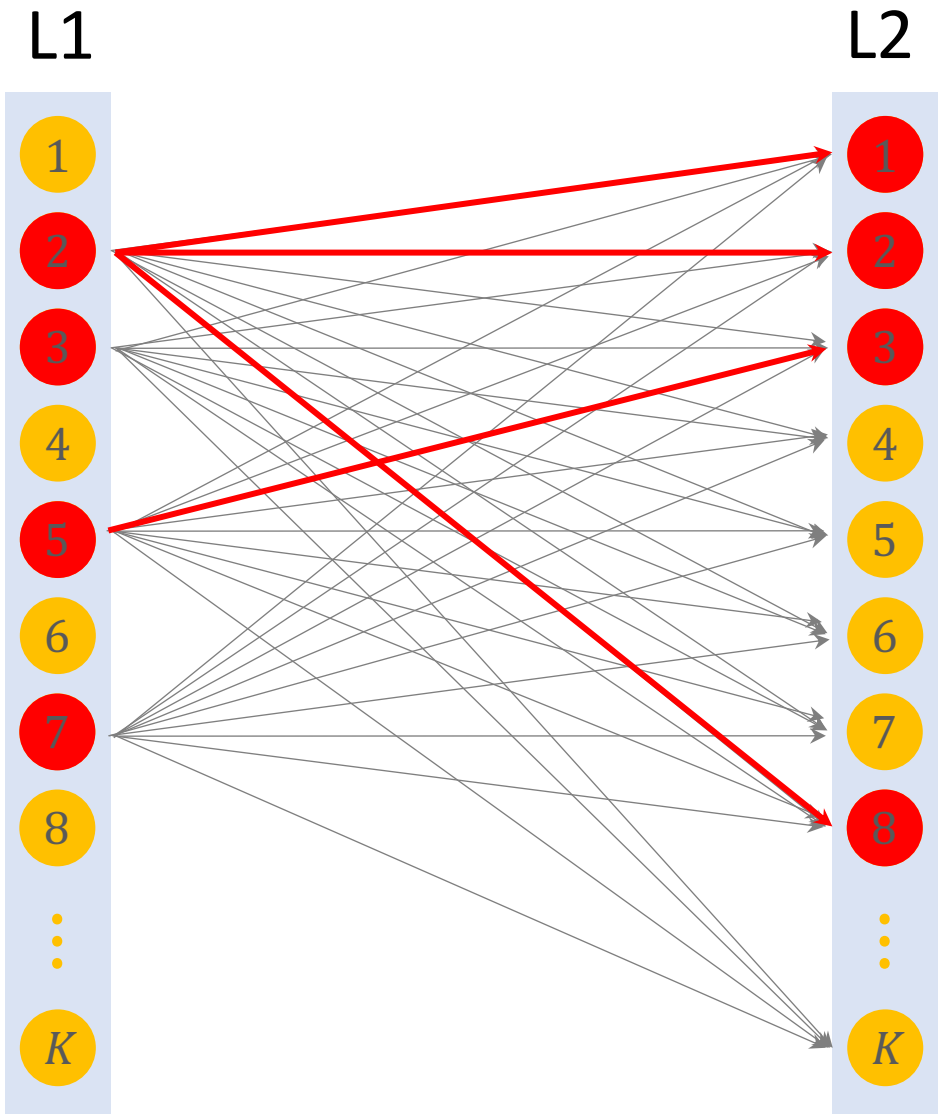


- 对于每个被选中的节点 a，计算用户对路径 [a, b] 的兴趣：

$$p_1(a|\mathbf{x}) \times p_2(b|a;\mathbf{x}).$$

- 算出 $4\times K$ 个分数，每个分数对应一条路径，选出分数 top 4 路径。

Beam Search (size = 4)

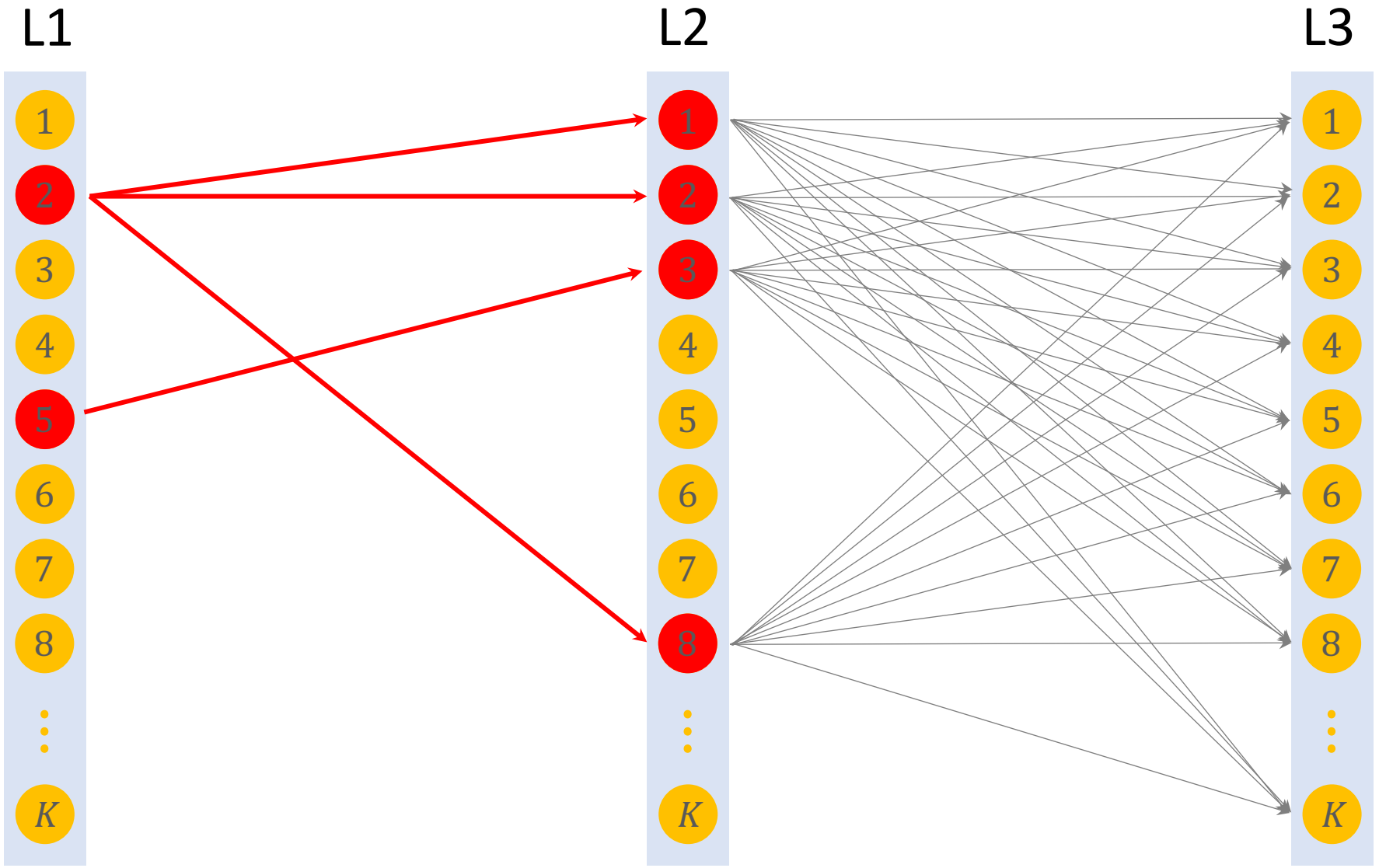


- 对于每个被选中的节点 a，计算用户对路径 [a, b] 的兴趣：

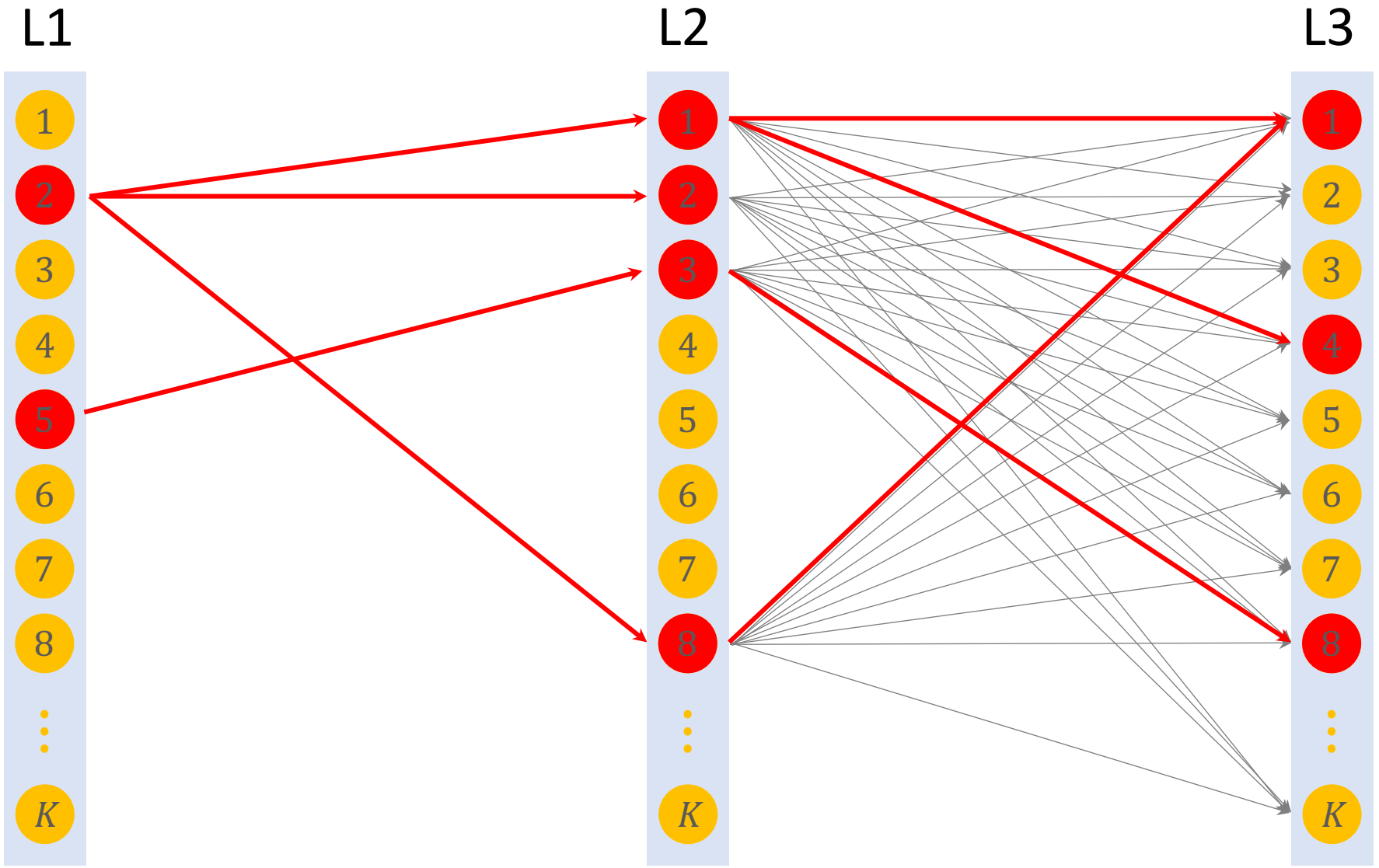
$$p_1(a|\mathbf{x}) \times p_2(b|a;\mathbf{x}).$$

- 算出 $4\times K$ 个分数，每个分数对应一条路径，选出分数 top 4 路径。

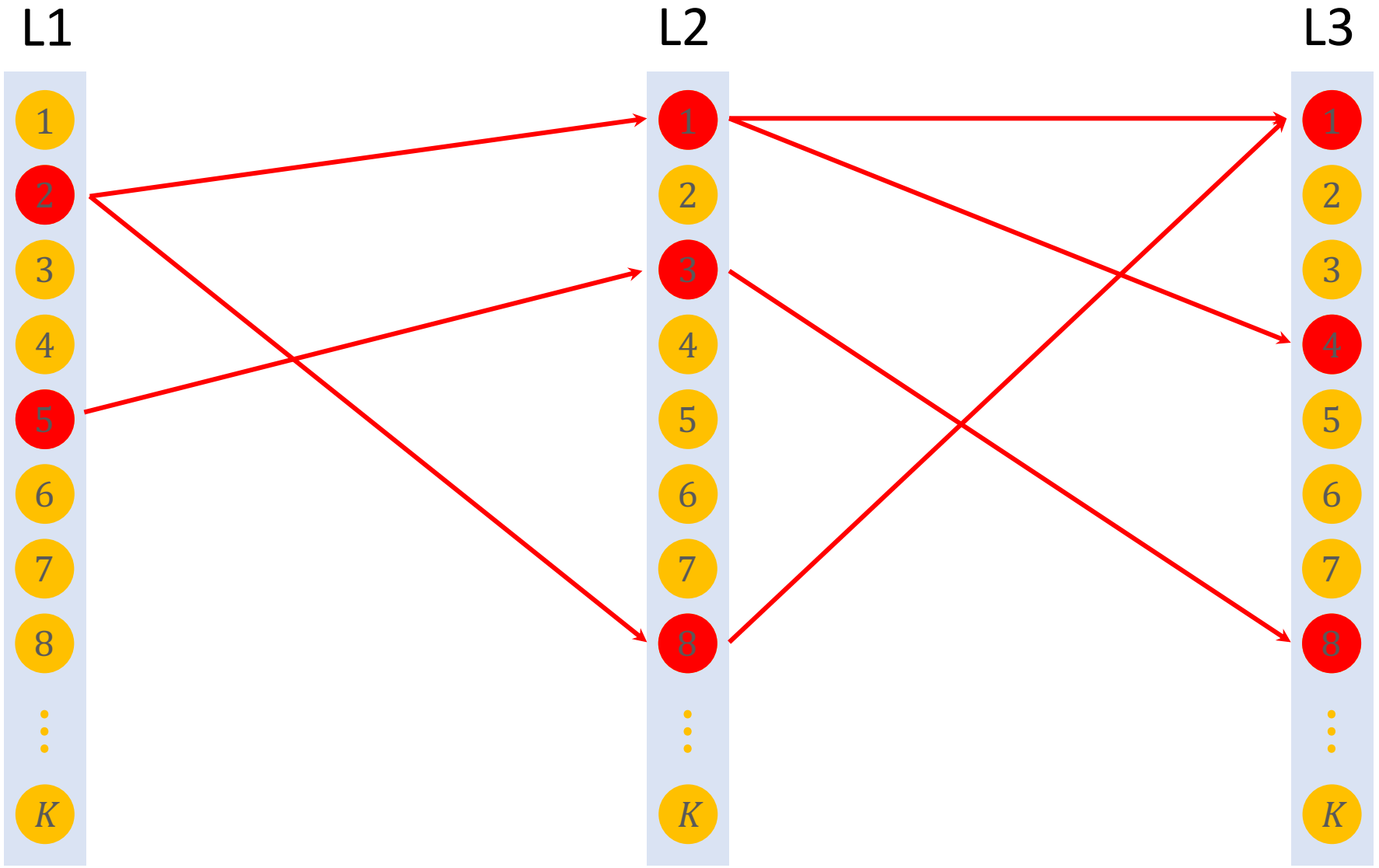
Beam Search (size = 4)



Beam Search (size = 4)



Beam Search (size = 4)



线上召回

- 第一步：给定用户特征，用神经网络做预估，用 beam search 召回一批路径。
- 第二步：利用索引，召回一批物品。
  - 查看索引 path → List⟨item⟩。
  - 每条路径对应多个物品。
- 第三步：对物品做排序，选出一个子集。

线上召回： user → path → item

训练

训练

同时学习神经网络参数和物品表征

- 神经网络 $p(a, b, c \mid \mathbf{x})$ 预估用户对路径 $[a, b, c]$ 的兴趣。
- 把一个物品表征为多条路径 $\{[a, b, c]\}$，建立索引：
  - item → List⟨path⟩，
  - path → List⟨item⟩。
- 正样本 (user, item)： click(user, item) = 1。

学习神经网络参数

- 物品表征为 J 条路径：$[a_1, b_1, c_1], \cdots, [a_J, b_J, c_J]$。
- 用户对路径 $[a, b, c]$ 的兴趣：

$$p(a, b, c \mid \mathbf{x}) = p_1(a \mid \mathbf{x}) \times p_2(b \mid a; \mathbf{x}) \times p_3(c \mid a, b; \mathbf{x}).$$

- 如果用户点击过物品，说明用户对 J 条路径感兴趣。

学习神经网络参数

- 物品表征为 J 条路径：$[a_1, b_1, c_1], \cdots, [a_J, b_J, c_J]$。
- 用户对路径 $[a, b, c]$ 的兴趣：

$$p(a, b, c \mid \mathbf{x}) = p_1(a \mid \mathbf{x}) \times p_2(b \mid a; \mathbf{x}) \times p_3(c \mid a, b; \mathbf{x}).$$

- 如果用户点击过物品，说明用户对 J 条路径感兴趣。
- 应该让 $\sum_{j=1}^{J} p(a_j, b_j, c_j \mid \mathbf{x})$ 变大。
- 损失函数：$\text{loss} = -\log\left(\sum_{j=1}^{J} p(a_j, b_j, c_j \mid \mathbf{x})\right)$。

学习物品表征

- 用户 user 对路径 path $= [a, b, c]$的兴趣记作：

$$p(\text{path} \mid \text{user}) = p(a, b, c \mid \mathbf{x}).$$

- 物品 item 与路径 path 的相关性：

$$\text{score}(\text{item}, \text{path}) = \sum_{\text{user}} p(\text{path} \mid \text{user}) \times \text{click}(\text{user}, \text{item}).$$

用户对路径的兴趣　　是否点击 (0或1)

学习物品表征

- 用户 user 对路径 path $= [a, b, c]$的兴趣记作：

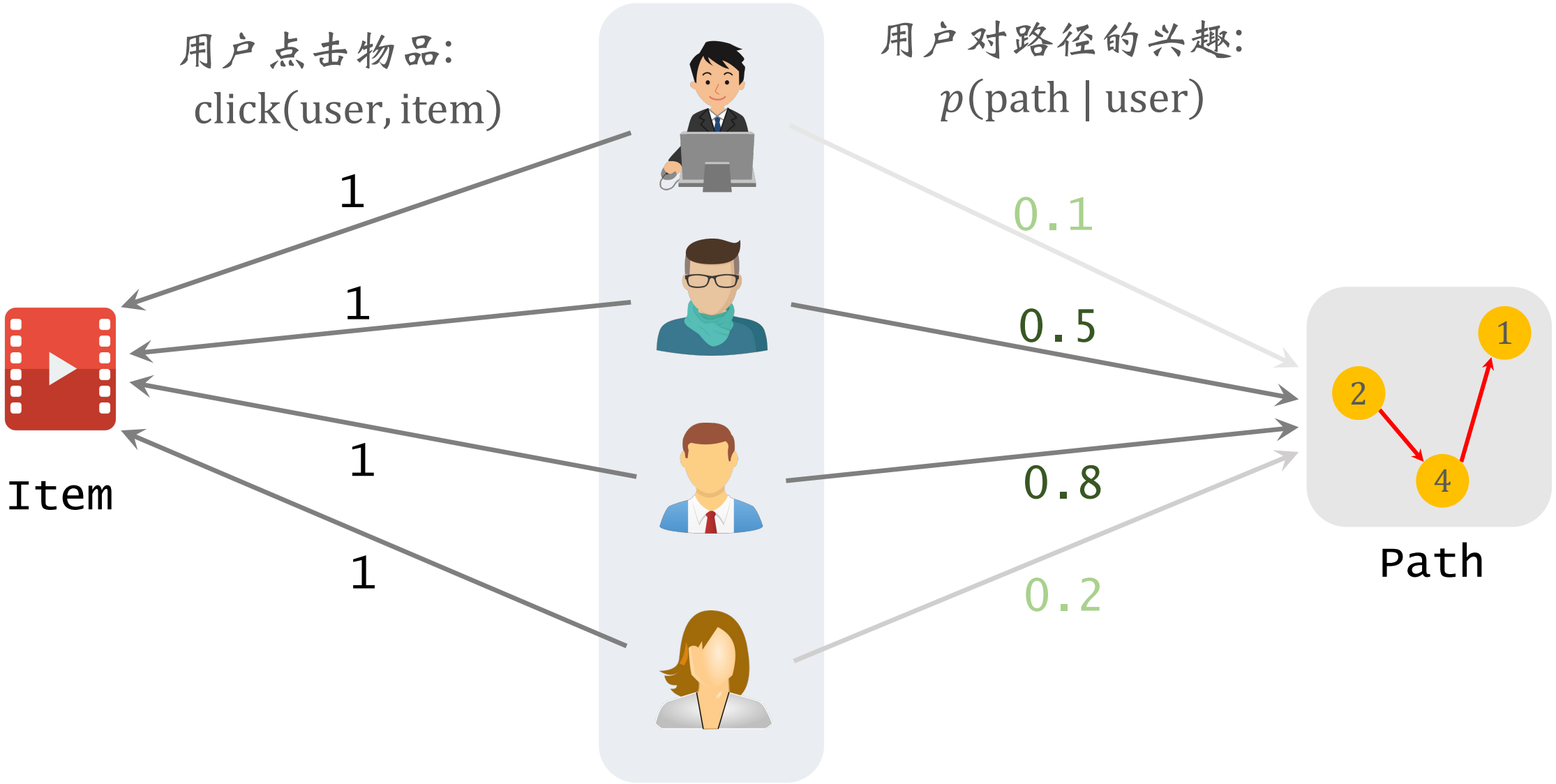
$$p(\text{path} \mid \text{user}) \ = \ p(a, b, c \mid \mathbf{x}).$$

- 物品 item 与路径 path 的相关性：

$$\text{score}(\text{item}, \text{path}) \ = \sum_{\text{user}} p(\text{path} \mid \text{user}) \times \text{click}(\text{user}, \text{item}).$$

- 根据 score(item, path) 选出 J 条路径作为 item 的表征。

表征：物品 → 路径


用户点击物品:
click(user, item)
用户对路径的兴趣:
$p(\text{path} \mid \text{user})$
1
1
1
1
0.1
0.5
0.8
0.2
Item
Path
1
2
4

学习物品表征

- 选出 J 条路径 $\Pi = \{\text{path}_1, \cdots, \text{path}_J\}$，作为物品的表征。
- 损失函数（选择与 item 高度相关的 path）：

$$\text{loss}(\text{item}, \Pi) = -\log\left(\sum_{j=1}^{J} \text{score}(\text{item}, \text{path}_j)\right).$$

- 正则项（避免过多的 item 集中在一条 path 上）：

$$\text{reg}(\text{path}_j) = \left(\text{number of items on path}_j\right)^4.$$

学习物品表征

用贪心算法更新路径

- 假设已经把物品表征为 J 条路径 $\Pi = \{\text{path}_1, \cdots, \text{path}_J\}$。
- 每次固定 $\{\text{path}_i\}_{i \neq l}$，并从未被选中的路径中，选出一条作为新的 path_l：

$$\text{path}_l \leftarrow \text{argmin}_{\text{path}_l} \text{loss}(\text{item}, \Pi) + \alpha \cdot \text{reg}(\text{path}_l).$$

- 选中的路径有较高的分数 $\text{score}(\text{item}, \text{path}_l)$，而且路径上的物品数量不会太多。

训练

更新神经网络

- 神经网络判断用户对路径的兴趣：

$$p(\text{path} \mid \mathbf{x}).$$

- 训练所需的数据：（1）“物品 → 路径”的索引，（2）用户点击过的物品。

- 如果用户点击过物品，且物品对应路径 path，则更新神经网络参数使 $p(\text{path} \mid \mathbf{x})$ 变大。

更新物品的表征

训练

更新神经网络

- 神经网络判断用户对路径的兴趣：

$$p(\text{path} \mid \mathbf{x}).$$

- 训练所需的数据：（1）“物品 → 路径”的索引，（2）用户点击过的物品。

- 如果用户点击过物品，且物品对应路径 path，则更新神经网络参数使 $p(\text{path} \mid \mathbf{x})$ 变大。

更新物品的表征

- 判断物品与路径的相关性：

物品 ⟵ 用户 ⟶ 路径

用户点击过物品　　神经网络的打分

- 让每个物品关联 J 条路径。
  - 物品和路径要有很高的相关性。
  - 一条路径上不能有过多的物品。

总结

Deep Retrieval

召回：用户 → 路径 → 物品

- 给定用户特征 $\boldsymbol{x}$，用神经网络预估用户对路径 $\text{path} = [a, b, c]$ 的兴趣，分数记作 $p(\text{path} \mid \mathbf{x})$.
- 用 beam search 寻找分数 $p(\text{path} \mid \mathbf{x})$ 最高的 s 条 path 。

Deep Retrieval

召回： 用户 → 路径 → 物品

- 给定用户特征 $\boldsymbol{x}$，用神经网络预估用户对路径 $\text{path} = [a, b, c]$ 的兴趣，分数记作 $p(\text{path} \mid \mathbf{x})$.
- 用 beam search 寻找分数 $p(\text{path} \mid \mathbf{x})$ 最高的 s 条 path 。
- 利用索引“ path → List⟨item⟩”召回每条路径上的 n 个物品。
- 一共召回 $s \times n$ 个物品，对物品做初步排序，返回分数最高的若干物品。

Deep Retrieval

训练： 同时学习 用户—路径 和 物品—路径 的关系

- 一个物品被表征为 J 条路径： path_1, …, path_J。
- 如果用户点击过物品，则更新神经网络参数，使分数增大：

$$\sum_{j=1}^{J} p(\text{path}_j \mid \mathbf{x}) .$$

Deep Retrieval

训练： 同时学习 用户—路径 和 物品—路径 的关系

- 一个物品被表征为 J 条路径： path_1, …, path_J。
- 如果用户点击过物品，则更新神经网络参数，使分数增大：

$$\sum_{j=1}^{J} p(\text{path}_j \mid \mathbf{x}) .$$

- 如果用户对路径的兴趣分数 $p(\text{path} \mid \mathbf{x})$ 较高，且用户点击过物品 item，则 item 与 path 具有相关性。
- 寻找与 item 最相关的 J 条 path，且避免一条路径上物品过多。

Thank You!

http://wangshusen.github.io/