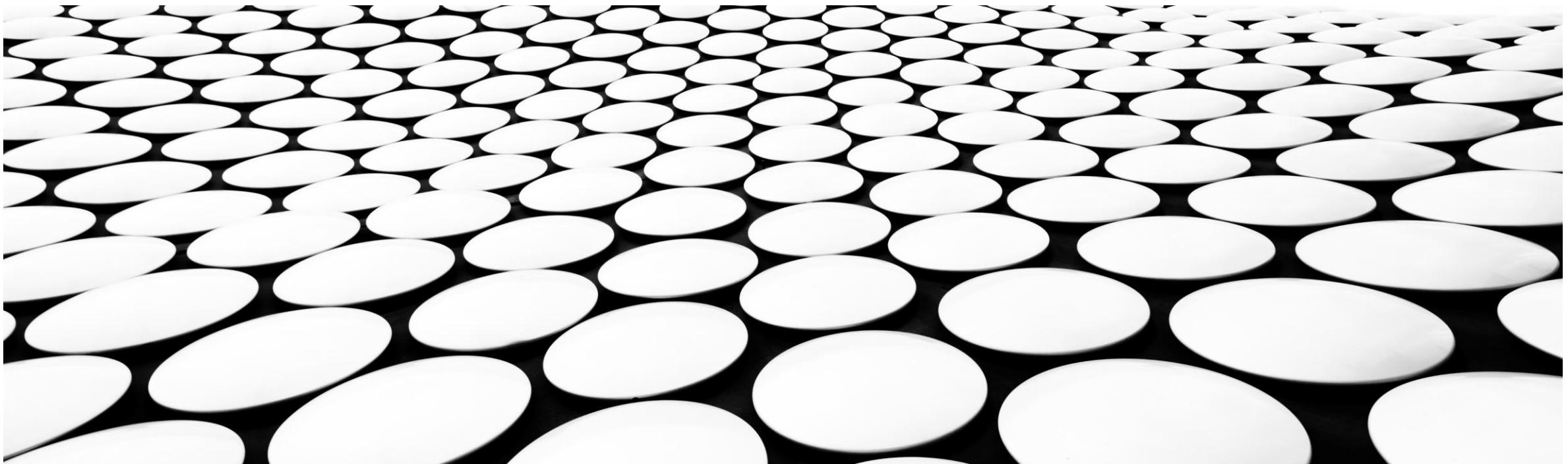

深度学习

邱怡轩



今天的主题

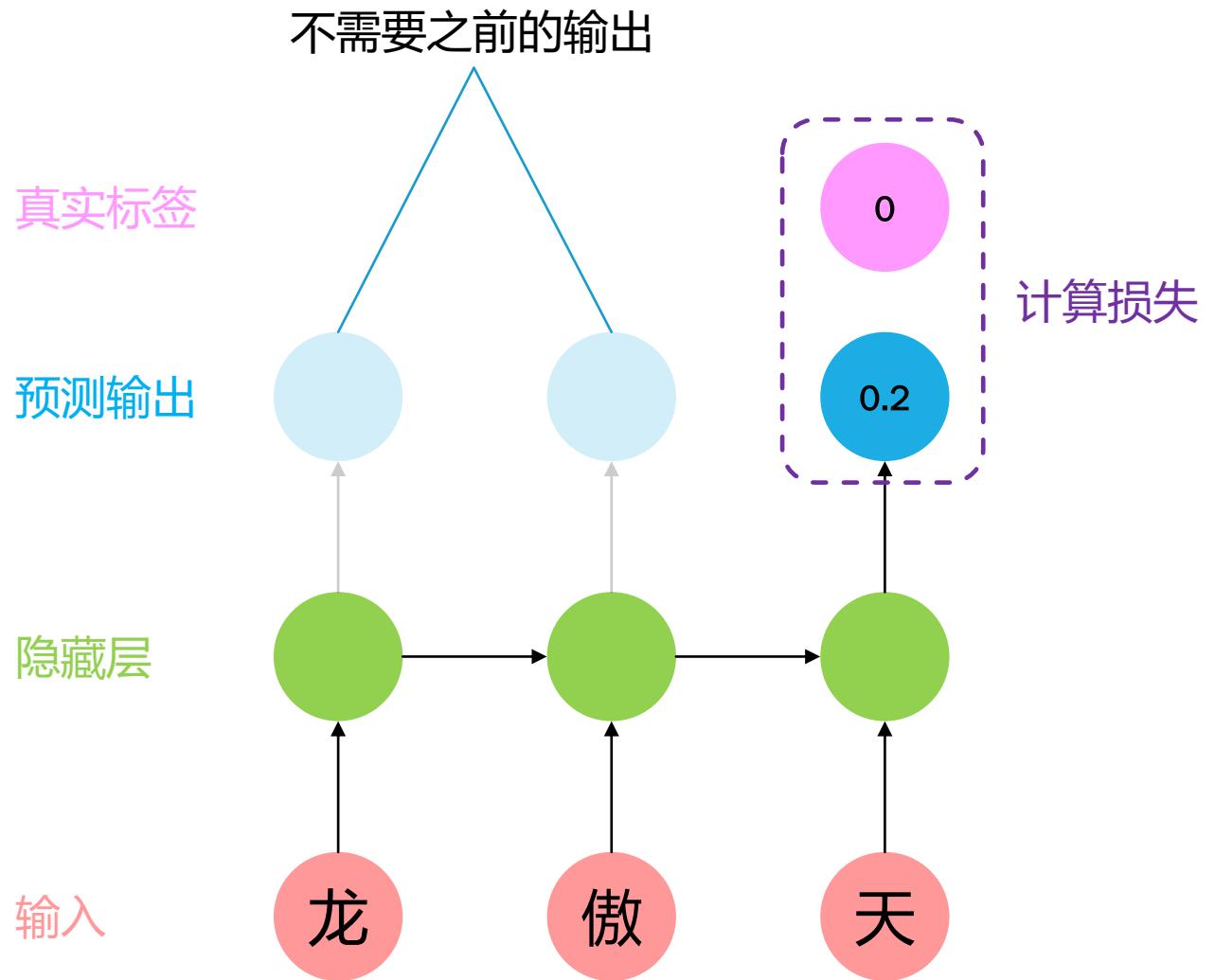
- RNN 生成模型 (续)



原理

回顾

■ 分类问题 (性别预测)



回顾

- 分类问题 (性别预测)
- 本质上是对 $p(y|x_1, \dots, x_T)$ 进行建模
- 这是一个条件分布
- Y 是标签, 视作服从 Bernoulli 分布
- $P(Y = 1)$ 的取值依赖于序列 x_1, \dots, x_T

生成模型

- 而序列生成模型是对 $p(x_1, \dots, x_T)$ 进行建模
- 这是一个联合分布
- 每个 X_t 代表一个字的 one-hot 向量
- 如何利用 RNN 对联合分布进行建模？

核心思想

- 核心思想依然是极大似然估计
- 条件概率分解公式
$$p(x_1, \dots, x_T) = p(x_1)p(x_2|x_1)p(x_3|x_1, x_2) \cdots p(x_T|x_1, \dots, x_{T-1})$$
- 事实上，这也是当前许多大语言模型的根本出发点
- 它表明，联合分布的建模可以拆解成一系列的序列分类问题

生成模型

- 人名生成可以看作是序列预测序列的问题
- 例如如下观测数据

张三
小五郎
福尔摩斯

- 可以转换成序列预测问题

自变量	因变量
[张, 三]	[三, <End>]
[小, 五, 郎]	[五, 郎, <End>]
[福, 尔, 摩, 斯]	[尔, 摩, 斯, <End>]

- <End>是特殊字符，表示名字的结尾

序列预测

- 更进一步，序列预测可以拆解成一系列分类问题

自变量

[小, 五, 郎]

因变量

[五, 郎, <End>]



自变量

[小]

[小, 五]

[小, 五, 郎]

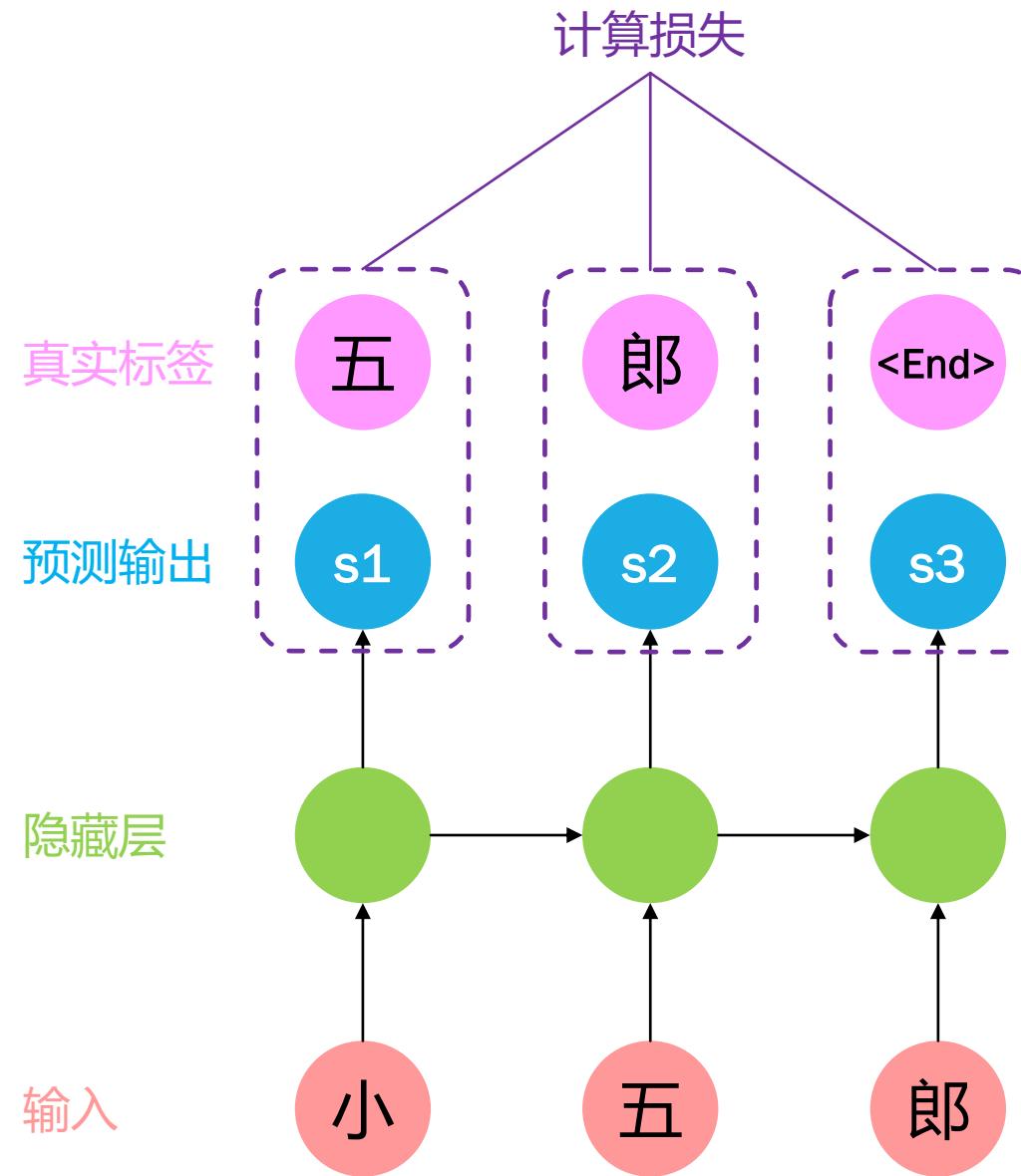
因变量

[五]

[郎]

[<End>]

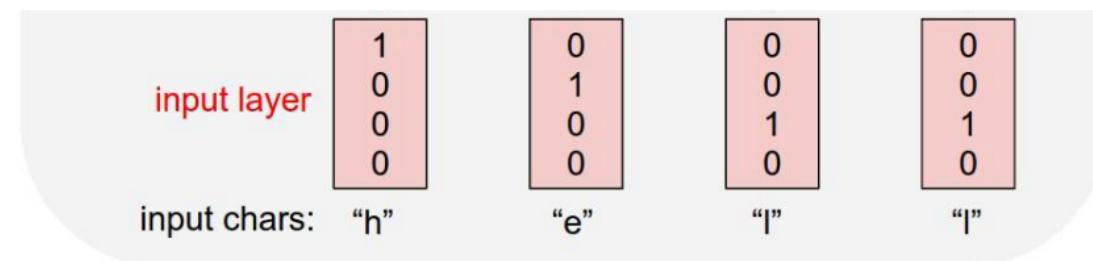
模型原理



Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

Example training
sequence:
“hello”

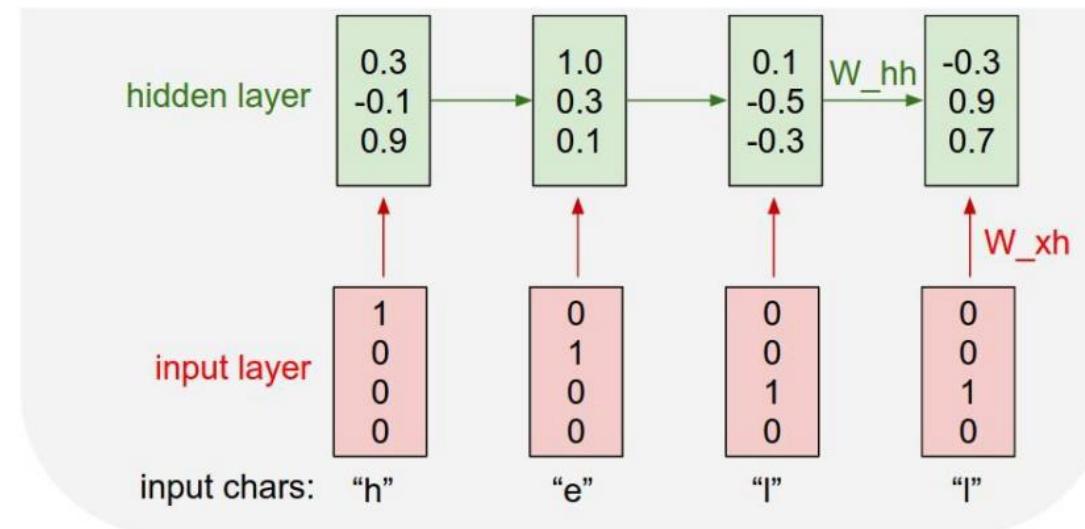


Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

Example training
sequence:
“hello”

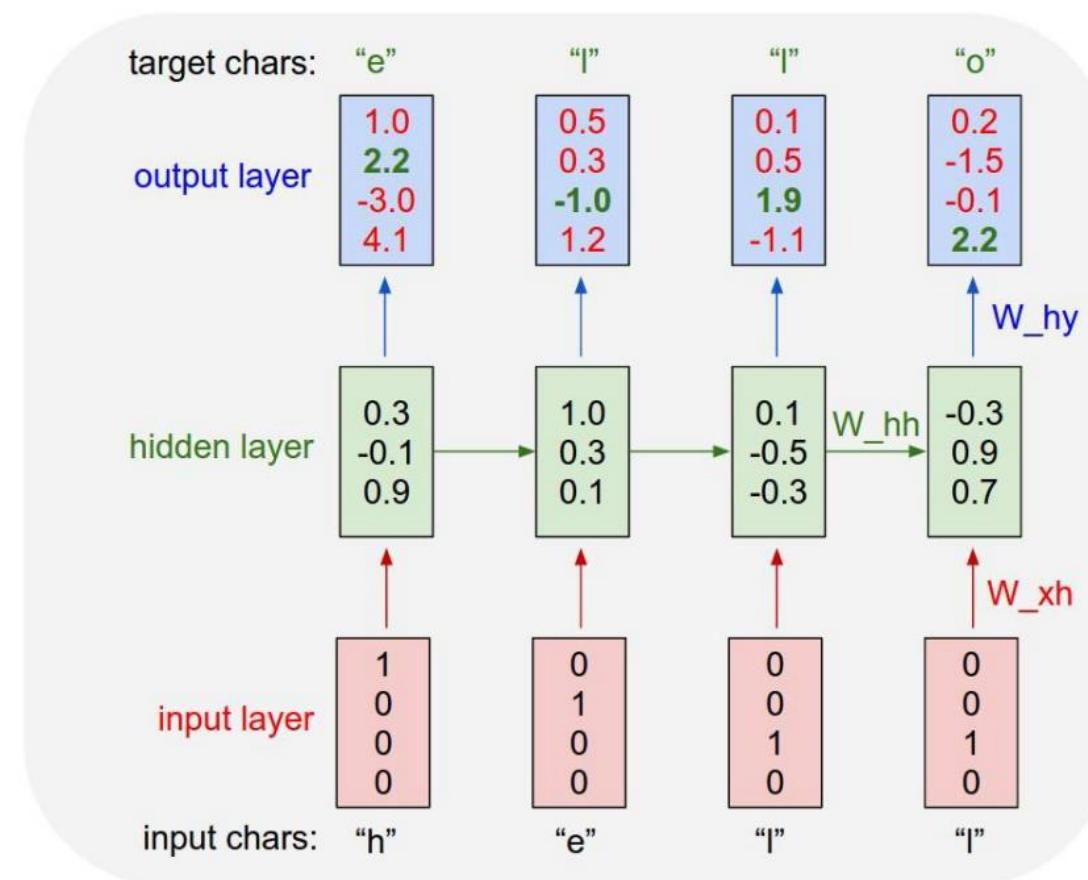
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

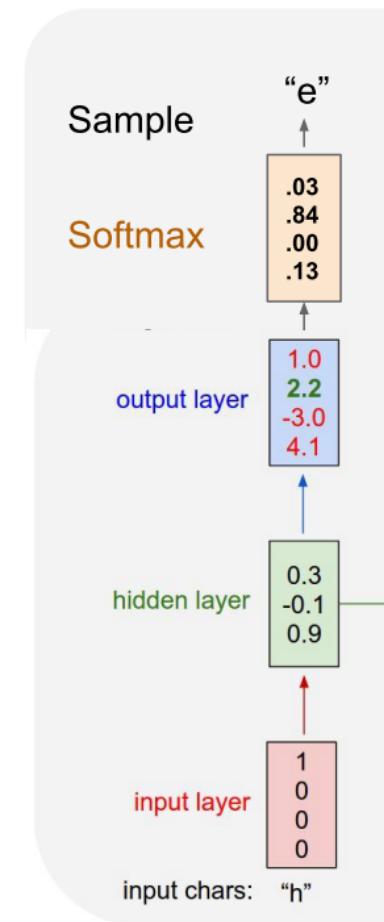
Example training
sequence:
“hello”



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

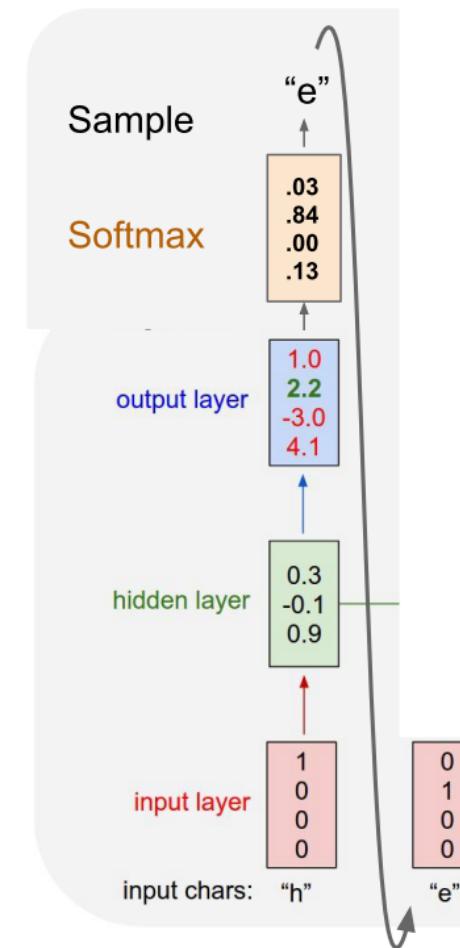
At test-time sample
characters one at a time,
feed back to model



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

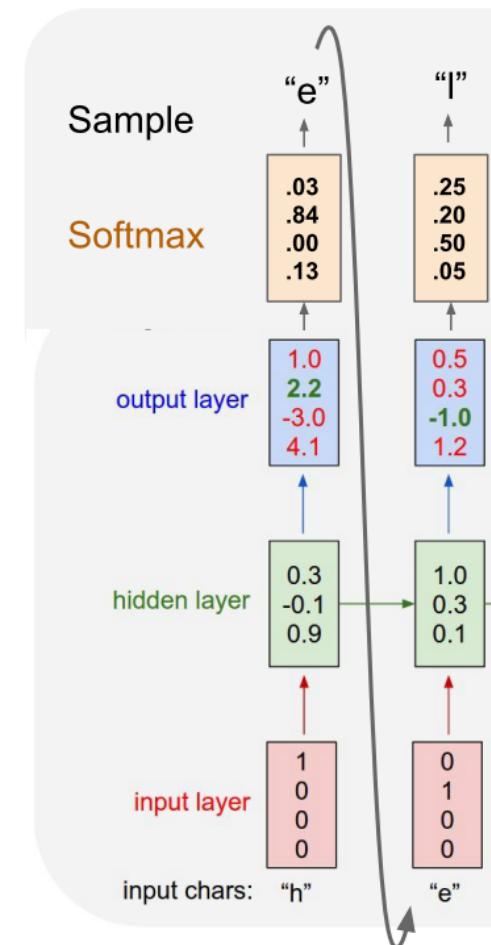
At test-time sample
characters one at a time,
feed back to model



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

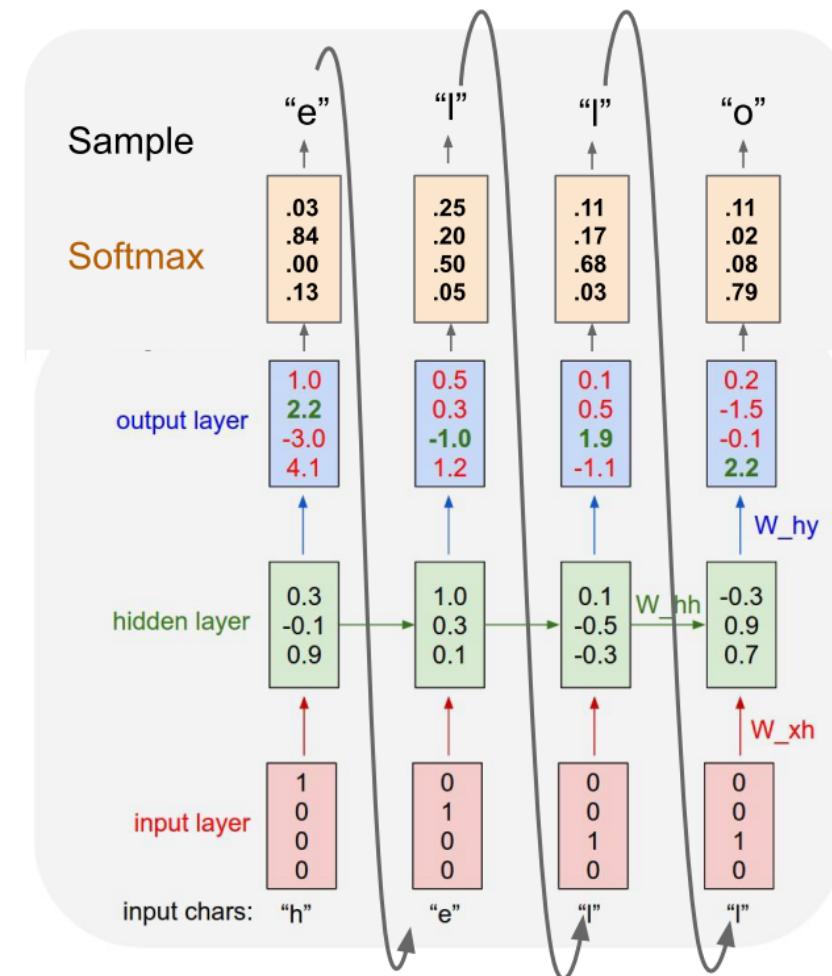
At test-time sample
characters one at a time,
feed back to model



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

At test-time sample
characters one at a time,
feed back to model



实例演示

人名生成

- 数据来源
- <https://github.com/wainshine/Chinese-Names-Corpus>
- >42万条英文翻译人名

5 阿阿巴伊	92901 帕多奈小	144332 海尔波莱宁	221388 劳豆斯基	319237 萨布多维奇
6 阿阿贝伊奥卢	92962 德罗茨	144333 海尔博恩	221389 劳登斯卡	319238 萨蒂纳拉扬
7 阿阿比金	92963 德罗代	144334 海尔博伊	221390 劳迪	319239 萨蒂南
8 阿阿尔坦	92964 德罗德	144335 海尔布科	221391 劳迪安斯基	319240 萨蒂宁
9 阿阿吉尔	92965 德罗德洛	144336 海尔布龙	221392 劳迪奥	319241 萨蒂农德
10 阿阿卡伊	92966 德罗迪	144337 海尔布龙纳	221393 劳迪德	319242 萨蒂诺
11 阿阿拉尔	92967 德罗蒂	144338 海尔布伦	221394 劳迪尔	319243 萨蒂皮
12 阿阿兰坦	92968 德罗尔	144339 海尔布特	221395 劳迪弗	319244 萨蒂什
13 阿阿姆恰姆	92969 德罗尔克	144340 海尔采	221396 劳迪迈茨基	319245 萨蒂斯
14 阿阿内尔	92970 德罗尔斯哈根	144341 海尔采尔	221397 劳迪诺	319246 萨蒂亚
15 阿阿奇迪肯	92971 德罗法	144342 海尔采戈夫	221398 劳迪奇	319247 萨蒂亚查兰
16 阿阿奇汗	92972 德罗夫	144343 海尔采戈娃	221399 劳迪奇纳	319248 萨蒂亚尔
17 阿阿恰亚克	92973 德罗夫斯基	144344 海尔采戈瓦茨	221400 劳迪萨	319249 萨蒂亚吉特
18 阿阿韦尔迪	92974 德罗盖	144345 海尔采格	221401 劳迪斯	319250 萨蒂亚凯图
19 阿阿哲克	92975 德罗盖奥	144346 海尔采格菲	221402 劳迪西	319251 萨蒂亚纳拉亚纳
20 阿阿哲克奥卢	92976 德罗盖蒂	144347 海尔采格福尔维	221403 劳迪西奥	319252 萨蒂亚纳拉扬
21 阿埃蒂斯	92977 德罗盖伊	144348 海尔采格哈齐	221404 劳迪希奇	319253 萨蒂亚纳拉因
22 阿埃多	92978 德罗戈	144349 海尔采格豪尔米	221405 劳迪亚	319254 萨蒂亚南达
23 阿埃尔	92979 德罗戈菲	144350 海尔采格利亚	221406 劳迪耶罗	319255 萨蒂亚南德
24 阿埃尔曼	92980 德罗戈雷亚努	144351 海尔采吉	221407 劳蒂	319256 萨蒂亚帕尔
25 阿埃费菜	92981 德罗戈什	144352 海尔采尼克	221408 劳蒂艾宁	319257 萨蒂亚普里亚
26 阿埃季	92982 德罗格	144353 海尔察	221409 劳蒂奥	319258 萨蒂延德拉
27 阿埃科博耶	92983 德罗格雷	144354 海尔茨	221410 劳蒂奥阿霍	319259 萨蒂扬
28 阿埃拉特尔	92984 德罗格里	144355 海尔茨卡	221411 劳蒂奥拉	319260 萨丁
29 阿埃莱内伊	92985 德罗格里亚	144356 海尔茨库	221412 劳蒂卡	319261 萨丁马
30 阿埃里略	92986 德罗格塞特	144357 海尔达尔	221413 劳蒂拉	319262 萨丁尼亚
31 阿埃利萨贝泰	92987 德罗格特	144358 海尔代洛	221414 劳蒂莫	319263 萨丁奇
32 阿埃利翁	92988 德罗古	144359 海尔代伊	221415 劳蒂齐	319264 萨乐
	92989 德罗古尔	144360 海尔当	221416 劳蒂西	319265 萨杜

预处理

- 出于演示目的
- 只考虑数据中最高频的50个字
- 建立字典

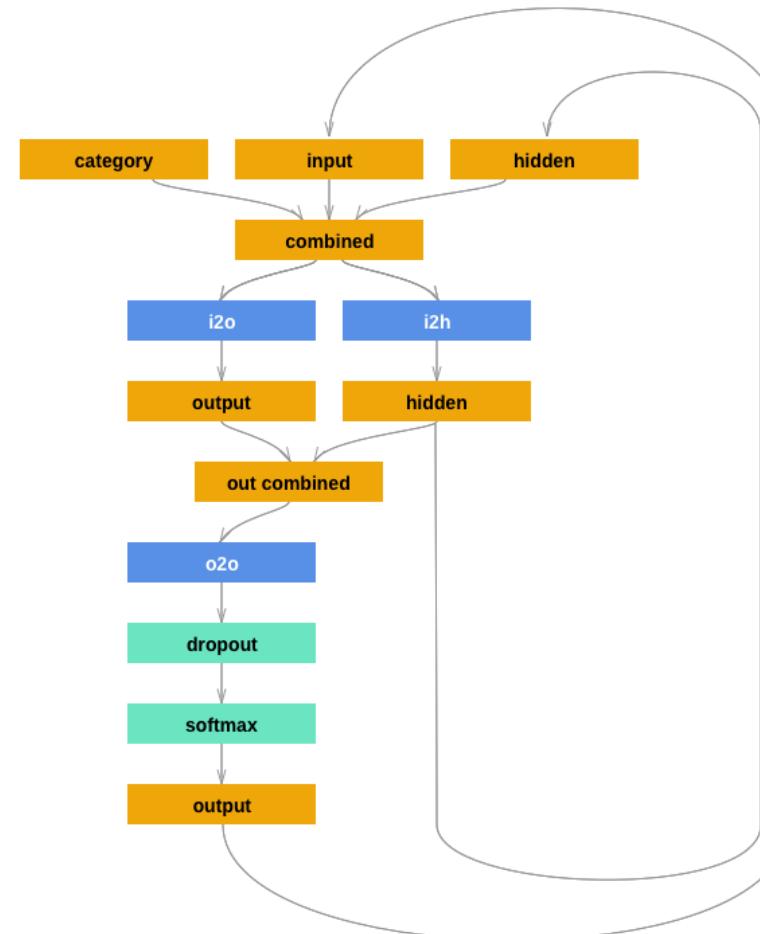
```
array(['尔', '斯', '拉', '克', '特', '德', '尼', '夫', '利', '科', '里', '卡', '亚',  
       '罗', '奇', '格', '布', '基', '维', '阿', '马', '伊', '奥', '纳', '雷', '洛',  
       '蒂', '塔', '诺', '瓦', '托', '莱', '巴', '内', '贝', '普', '萨', '娃', '耶',  
       '迪', '米', '西', '勒', '什', '达', '埃', '加', '库', '帕', '韦'], dtype='<U1')
```

- 还需要在字典中加入一个特殊字符代表字符串结束

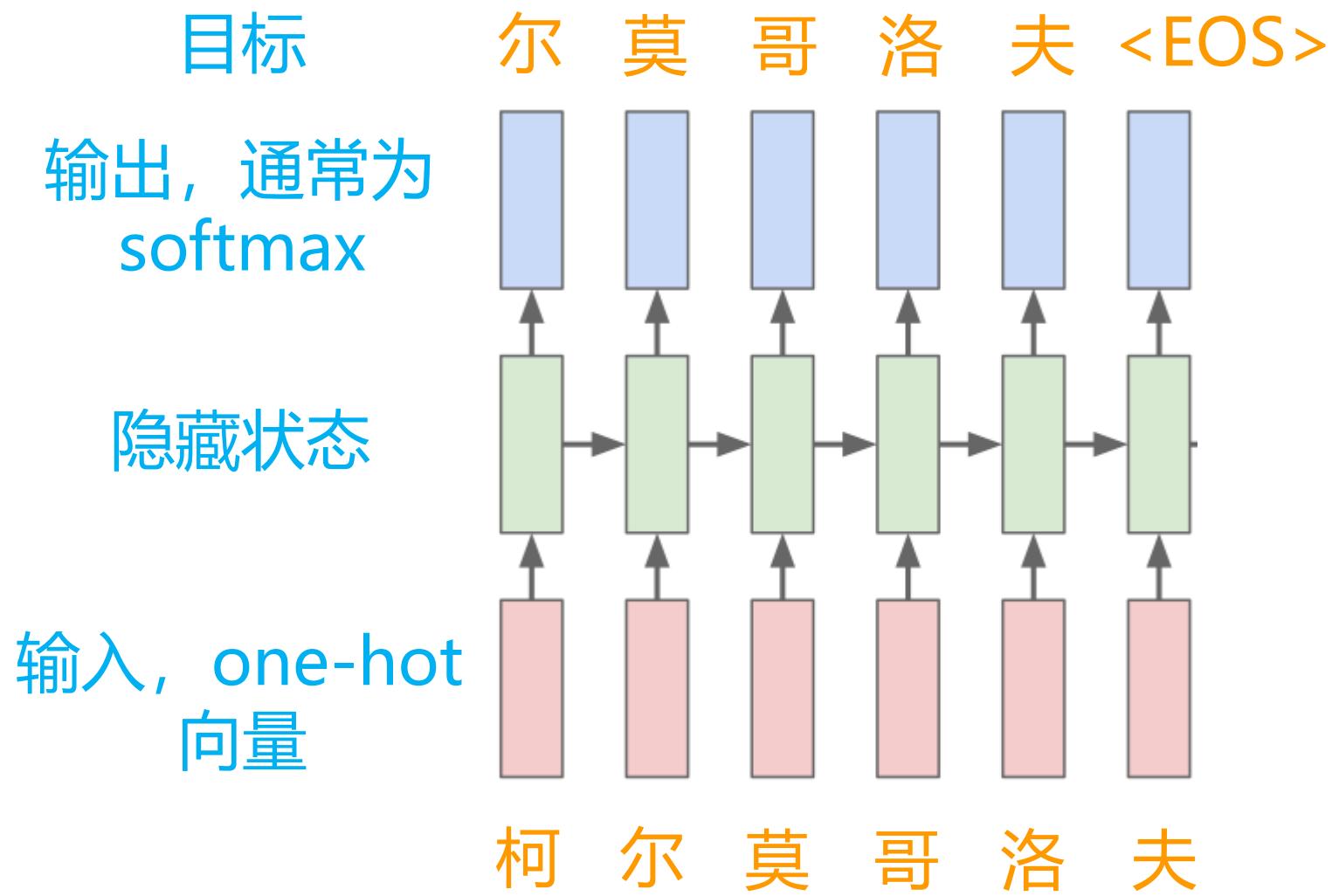
建立模型

■ 参考：

https://pytorch.org/tutorials/intermediate/char_rnn_generation_tutorial.html



损失函数



训练结果

- 比用序列预测分类更复杂
- GPU 上训练 ~12 分钟

```
[雷斯塔, 伊尔马斯, 瓦尔德, 耶尔维, 西尔韦斯特里, 布拉伊科夫斯基,  
库尔蒂耶, 迪亚科, 拉斯特里, 罗斯托, 维尔斯, 马尔基奥, 埃斯特拉德,  
维尔斯, 韦尔尼, 维尔斯, 马尔基奥, 格拉斯科, 库尔蒂耶, 尔德,  
萨尔瓦尼, 维尔斯, 巴尔巴里尼, 内斯托拉, 克拉斯尼奇, 伊尔马斯,  
德拉斯科, 诺尔贝格, 伊尔马斯, 德拉斯科, 马尔基奥, 勒布罗, 达尔马斯,  
莱斯科, 瓦尔德, 西尔韦斯特里, 罗斯托, 托尔托拉, 米尔科, 特里斯塔尼,  
耶尔维, 罗斯托, 奇科, 贝尔托利尼, 克拉斯尼奇, 迪亚科, 利亚尔迪,  
亚尔马, 塔尔迪, 耶尔维]
```

代码实现

- 参见 [lec12-rnn-generation.zip](#)