深度学习

机器也懂感情？

文本分类问题简介

张江

集智AI学园
campus.swarma.org

文本分类问题

- 当那四个女孩儿施暴夺去父亲生命时，她曾想冲上台去，但身边的两名老校工死死抓住她，并在耳边低声告诉她别连自己的命也不要了，当时会场已经处于彻底的癫狂，她的出现只会引出更多的暴徒。

- 在南宁的一个居民小区内，而且还是在一个幼儿托管机构里，出现了一只活生生的大鳄鱼，把孩子和老师都吓得够呛。

- Alice通过公开信道告诉Bob需要扔掉哪些错误测量结果，留下即为最终筛选结果，该结果和Alice和Bob端相同，即为量子秘钥

- 那么，按照他们的具体估计，那已被证明为位于临界线上的“无穷多个非平凡零点”跟全部非平凡零点相比，究竟占多大的百分比呢？答案可能沮丧得出乎读者们的意料：百分之零！

文本分类问题

网页分类

- 综合
- 游戏
- 科学
- ……

新闻分类

- 音乐
- 时事
- 娱乐
- ……

邮件分类

- 垃圾
- 广告
- 诈骗
- ……

个性化推荐

- 用户A
- 用户B
- 用户C
- ……

情绪分类

- 老熟客了,东西还是一如既往的好,货真价实的日货尾单,性价比突出.
- 好卖家，真有耐心，我终于买到想要的东西了。

- 这是黑店，都别买辣鸡中的战斗机，发来就是烂的。
- 面料薄的和纸一样，非常生气的一次购物。
- 一星都不想给，当时把包装袋随手扔了，现在退货还不知道退不退得了。

情绪分析

置顶 伊利股份暴涨！！我们都在声援马伊琍！！如果你也是个爱过的女人，如果你也痛恨硝小三，今天，请你购买伊利股份！！！ @Dia楚笛 @蓝鲸财经记者工作平台 @皎若云间月_u @财联社 @财经女记者部落

3月31日 09:53 来自微博 weibo.com (239) | 转发(2433) | 收藏 | 评论(397)

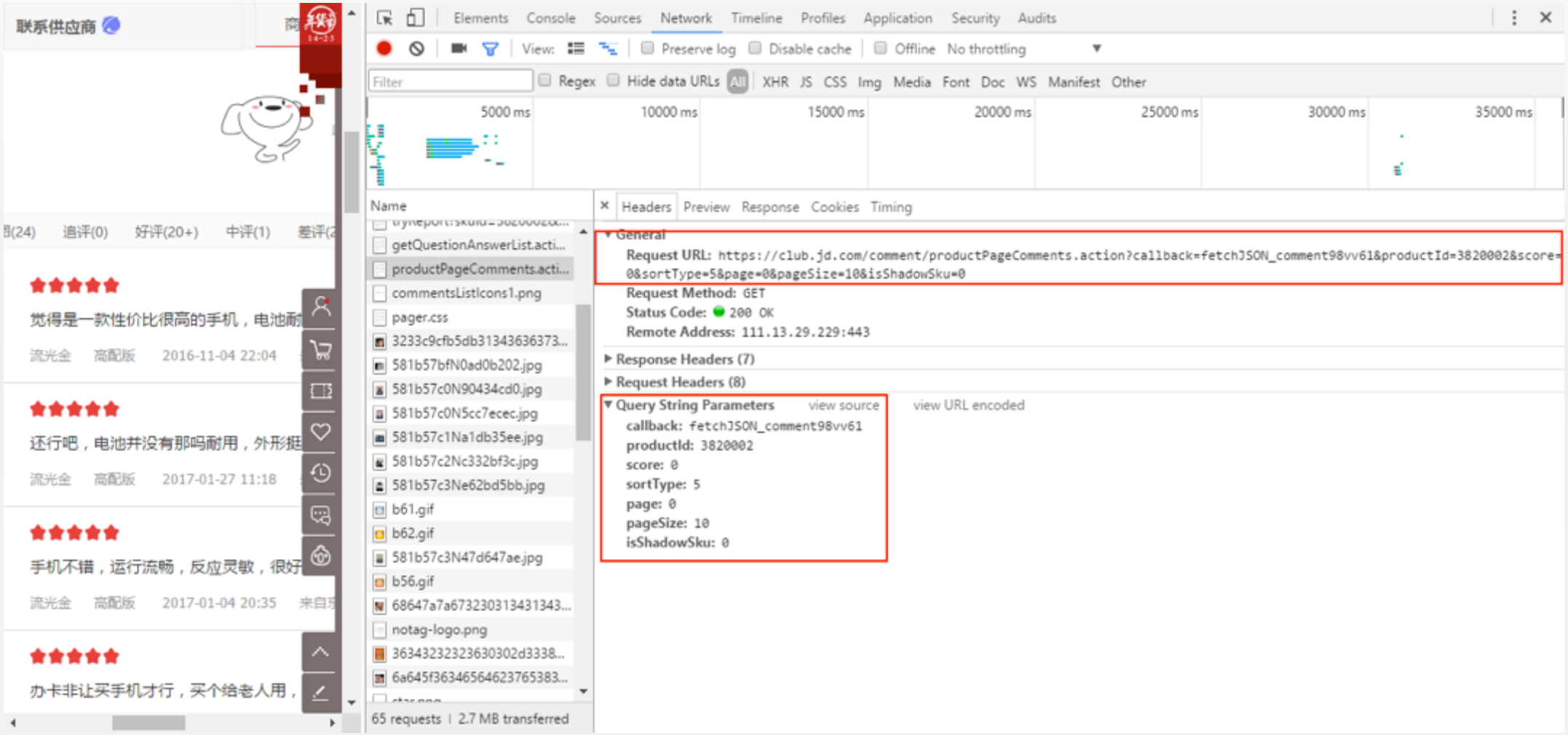
我们的任务

- 老熟客了,东西还是一如既往的好,货真价实的日货尾单,性价比突出.
- 好卖家，真有耐心，我终于买到想要的东西了。
- 无论款式和做工都棒棒嗒，老公很喜欢很喜欢了。谢谢掌柜。无论款式和做工都棒棒嗒，老公很喜欢服务好！礼品还真是漂亮，超值！但，有一颗小珠有些瑕疵，这是难免的

JD.COM 京东

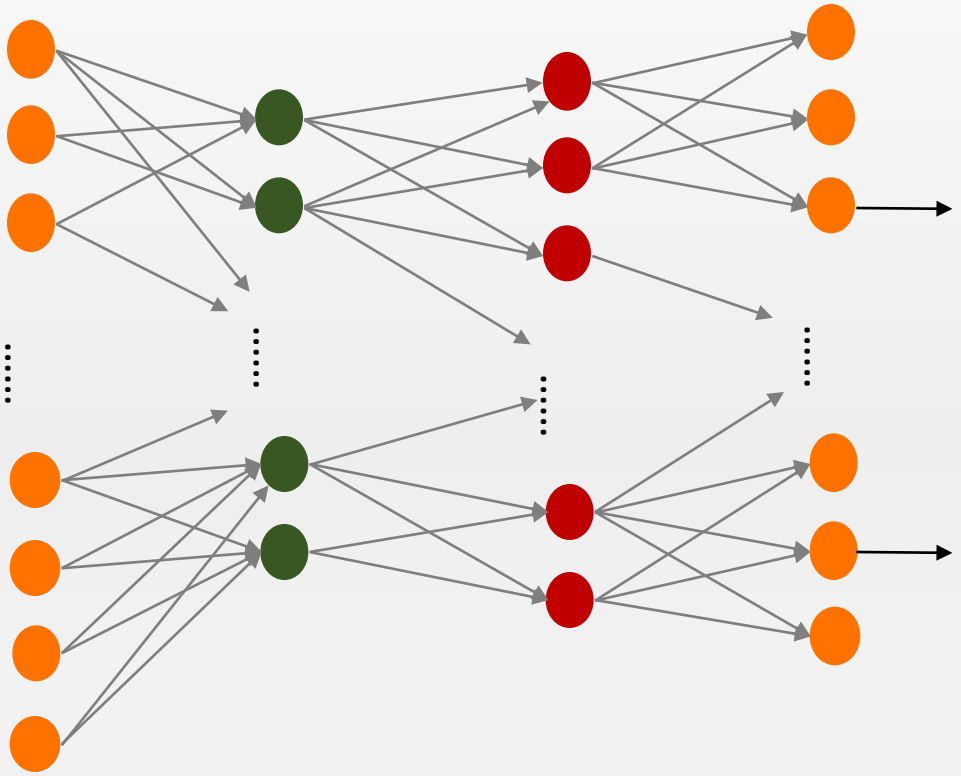
- 这是黑店，都别买辣鸡中的战斗机，发来就是烂的。
- 面料薄的和纸一样非常生气?的一次购物。
- 一星都不想给，当时把包装袋随手扔了，现在退货还不知道退不退得了。
- 真的好水的裤子，买了一天裤子有点紧，裤脚拉链那部分直接炸线了丑的要死
- 大家千万千万不要买 估计好评都是水军.一星都不想给。。。

数据获取


联系供应商
(24)
追评(0)
好评(20+)
中评(1)
流光金
高配版
2016-11-04 22:04
2017-01-27 11:18
2017-01-04 20:35
Elements
Console
Sources
Network
Timeline
Profiles
Application
Security
Audits
Preserve log
Disable cache
Offline
No throttling
Filter
Regex
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
5000 ms
10000 ms
15000 ms
20000 ms
25000 ms
30000 ms
35000 ms
Name
getQuestionAnswerList.acti...
productPageComments.acti...
commentsListIcons1.png
pager.css
581b57bfN0ad0b202.jpg
581b57c0N90434cd0.jpg
581b57c0N5cc7ecec.jpg
581b57c1Na1db35ee.jpg
581b57c2Nc332bf3c.jpg
581b57c3Ne62bd5bb.jpg
b61.gif
b62.gif
581b57c3N47d647ae.jpg
b56.gif
notag-logo.png
65 requests | 2.7 MB transferred
Headers
Preview
Response
Cookies
Timing
General
Request URL: https://club.jd.com/comment/productPageComments.action?callback=fetchJSON_comment98vv61&productId=3820002&score=0&sortType=5&page=0&pageSize=10&isShadowSku=0
Request Method: GET
Status Code: 200 OK
Remote Address: 111.13.29.229:443
Response Headers (7)
Request Headers (8)
Query String Parameters
view source
view URL encoded
callback: fetchJSON_comment98vv61
productId: 3820002
score: 0
sortType: 5
page: 0
pageSize: 10
isShadowSku: 0

处理文本信息的难点

- 文本的表示
  - 词向量?
  - 其它方式?
- 文本的长度是不固定的
  - 我们不能将所有的单词当作输入节点喂给神经网络
- 需要把文本向量化



词袋模型 (Bag of words)

词袋模型 (Bag of words)

- 将语料中的所有单词排序
- 所有语料中有多少词，向量就有多少维
- 对于一个句子，统计出每一个单词的出现次数，并将这个数字处理后填充到相应的位置，构成一个向量

词袋模型 (Bag of words)

- 我 爱 北京 天安门
- 每个 人 都有 一个 爱 的 人

单词表：
{我，爱，北京，天安门，每个，人，都有，一个，的}

我 爱 北京 天安门

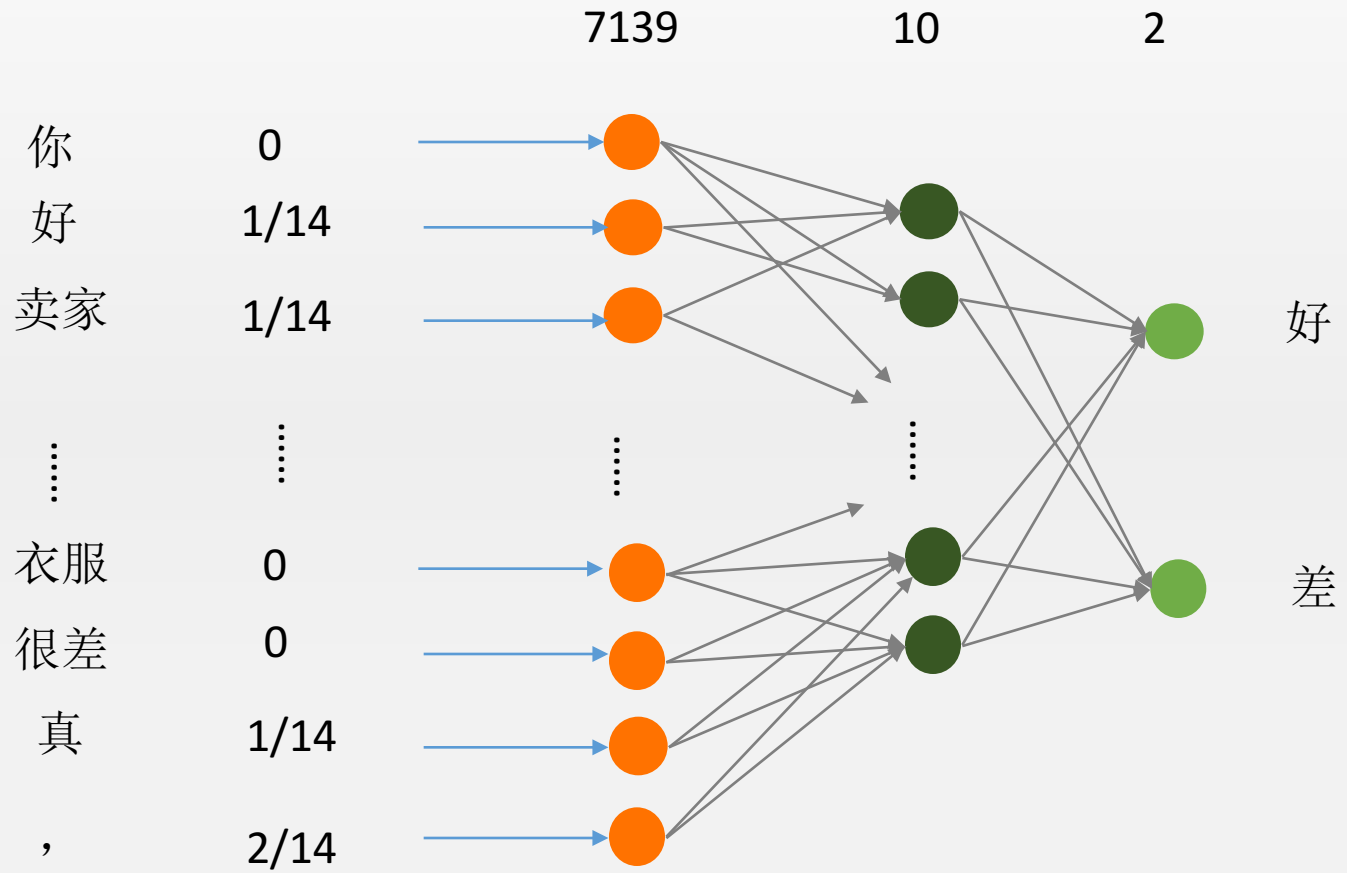
向量表示：
(1/4，1/4，1/4，1/4，0，0，0，0，0)

每个 人 都有 一个 爱 的 人

向量表示：
(0，1/7，0，0，1/7，2/7，1/7，1/7，1/7)

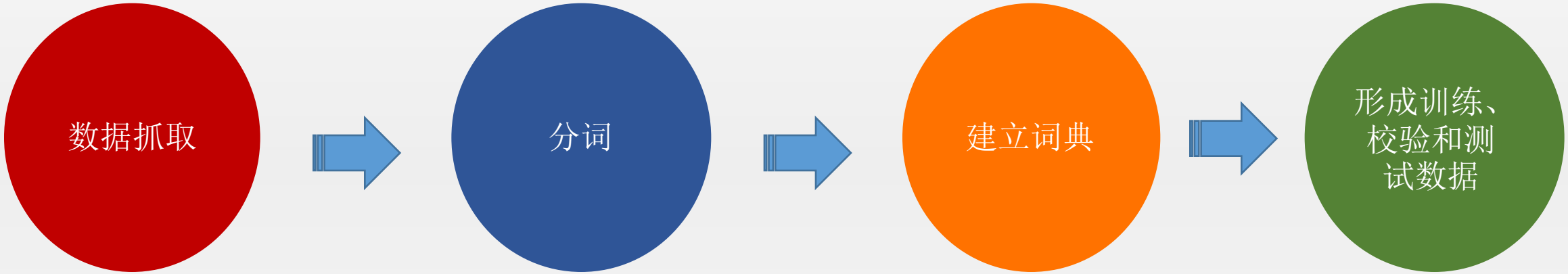
构造一个分类器

好 卖家，真 有 耐心，我 终于 买到 想要的 东西 了。



处理流程

准备数据



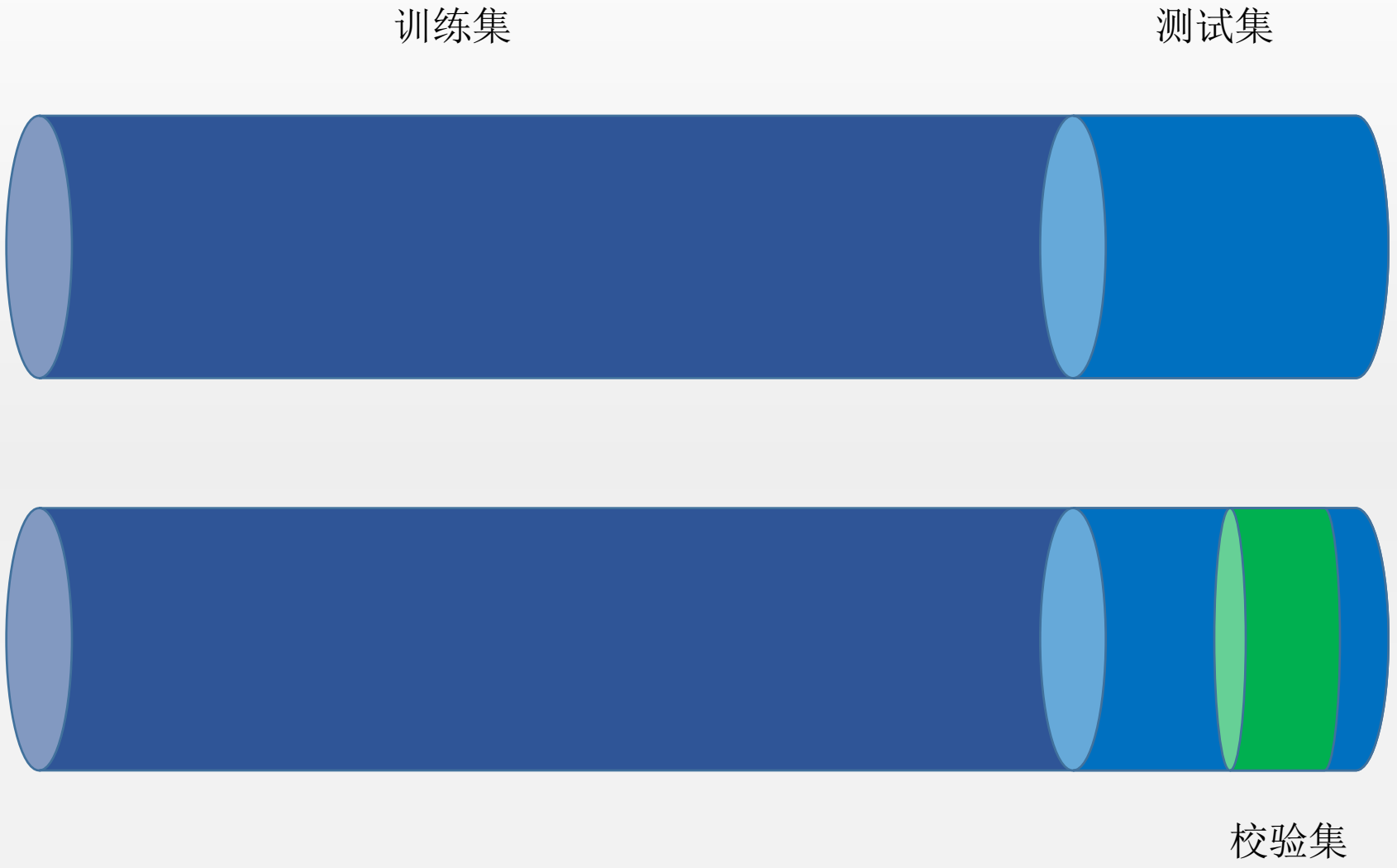
网页的抓取

- 京东的API
- Requests包
  - 让网页抓取更简洁
- Json包
  - JSON(JavaScript Object Notation, JS 对象标记) 是一种轻量级的数据交换格式。
  - 形如如下格式:
  - {"id":"1268900","name":"穿上很舒服","status":0,"rid":"11341","productId":10359162198,"count":3970,"modified":"2017-06-30 18:49:15","type":0,"canBeFiltered":false

建立字典

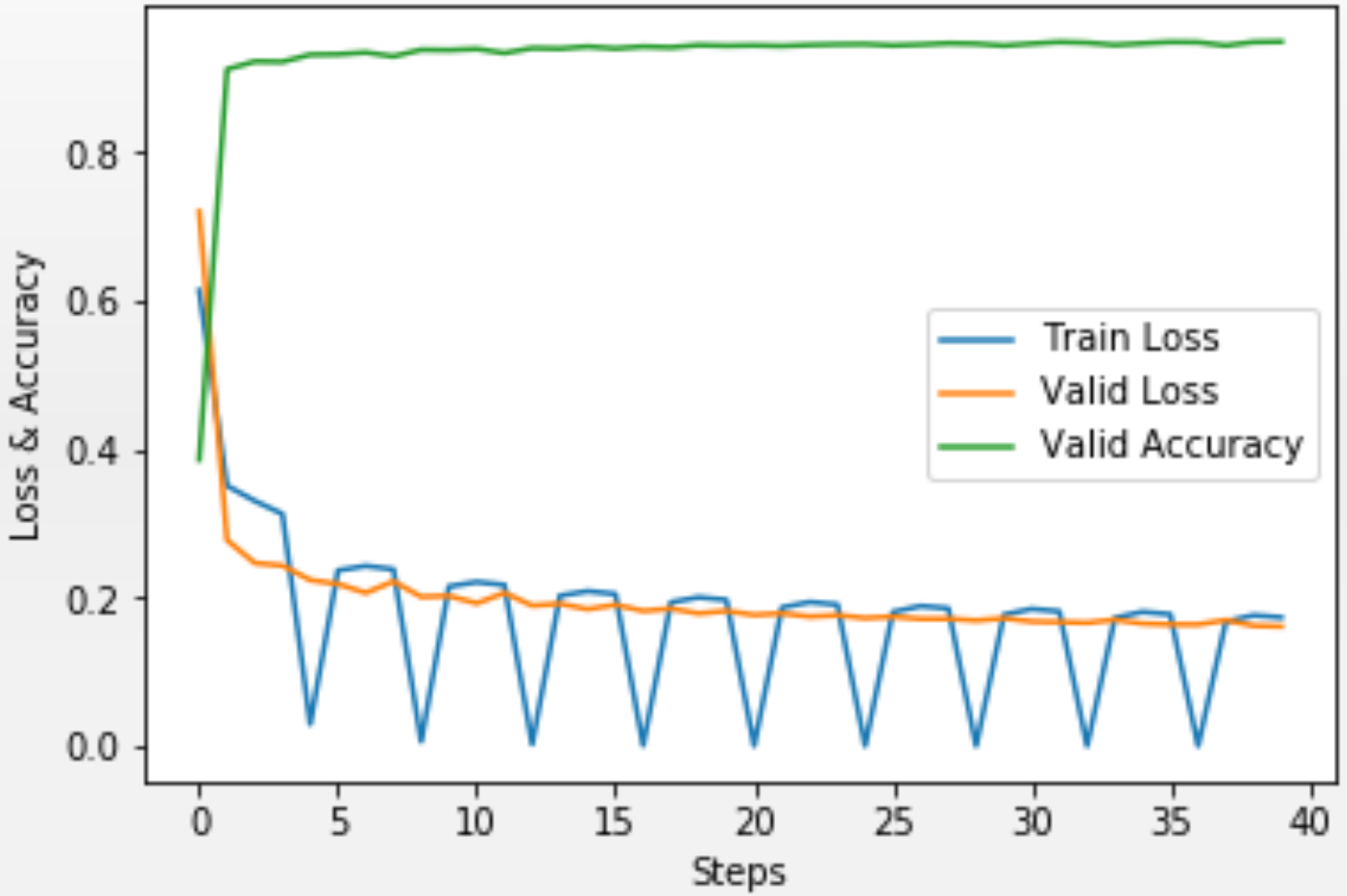
- Python中的字典:
- Dictionary是一种Python中的特殊存储结构，由（键，值）构成的集合
- 例如：diction = {‘name’:10, ‘height’:20, ‘width’:30,…}
- 所有元素的键不能重复，值可以重复
- 访问的时候，可以用diction[‘name’]快速访问
- 我们扫描所有的文本，为每一个单词建立了一个键值
  - {‘a’:[19,2], ‘about’:[0, 20], ‘the’:[1, 30]}

训练集（training）／校验集（validation/develop）／测试集（test）



训练集（training）／校验集（validation/develop）／测试集（test）

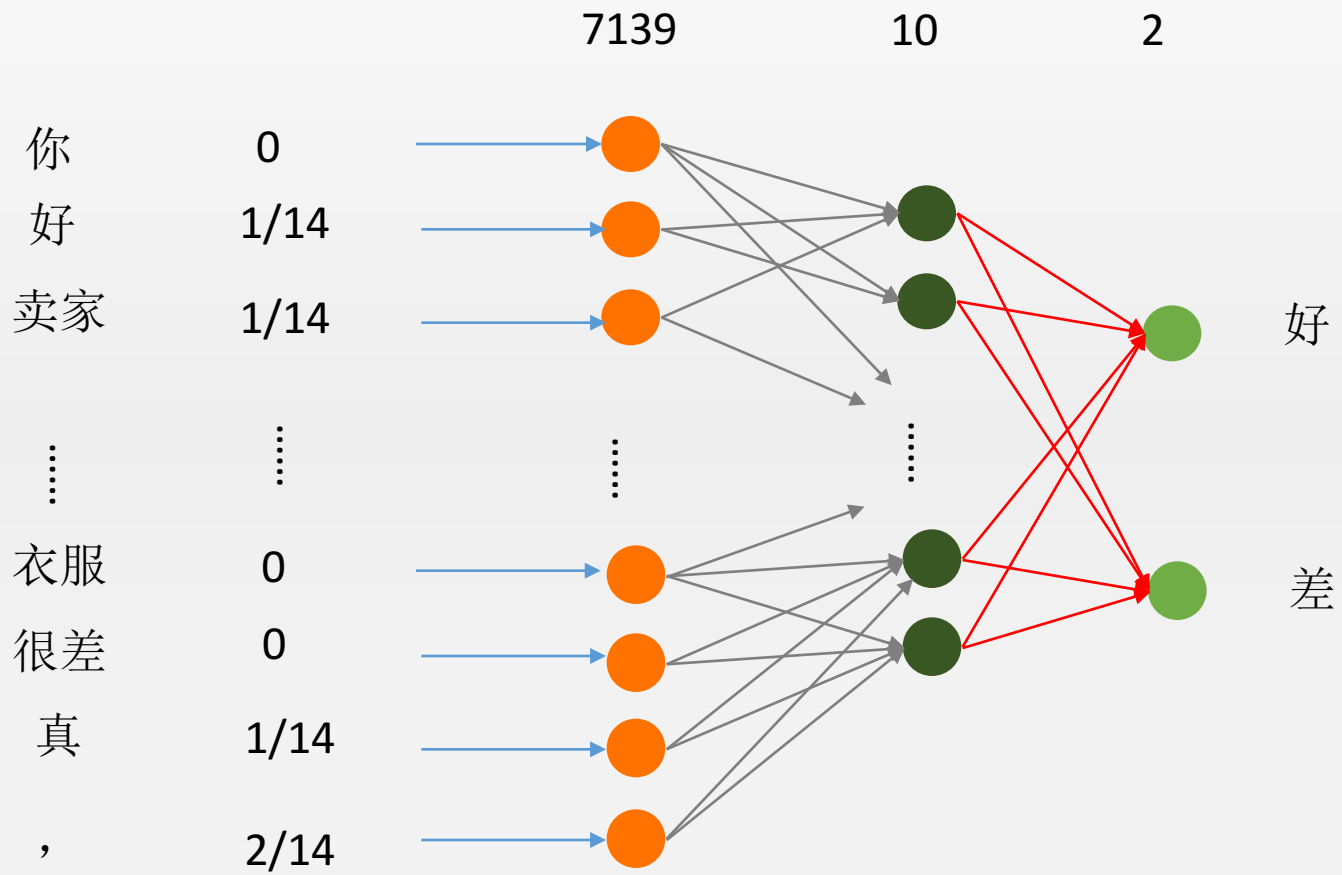
训练曲线



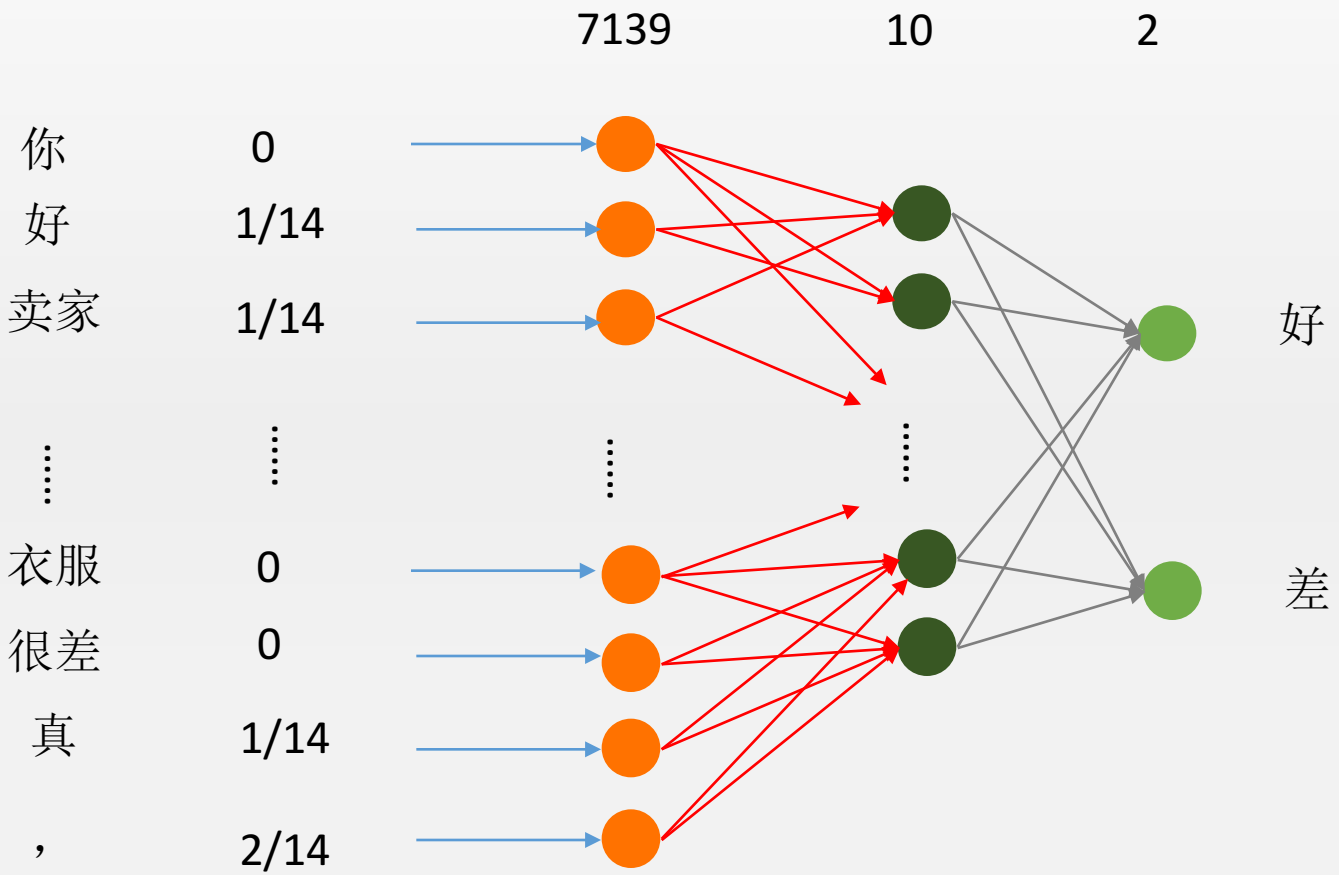
- 测试数据准确率：0.90

解剖神经网络

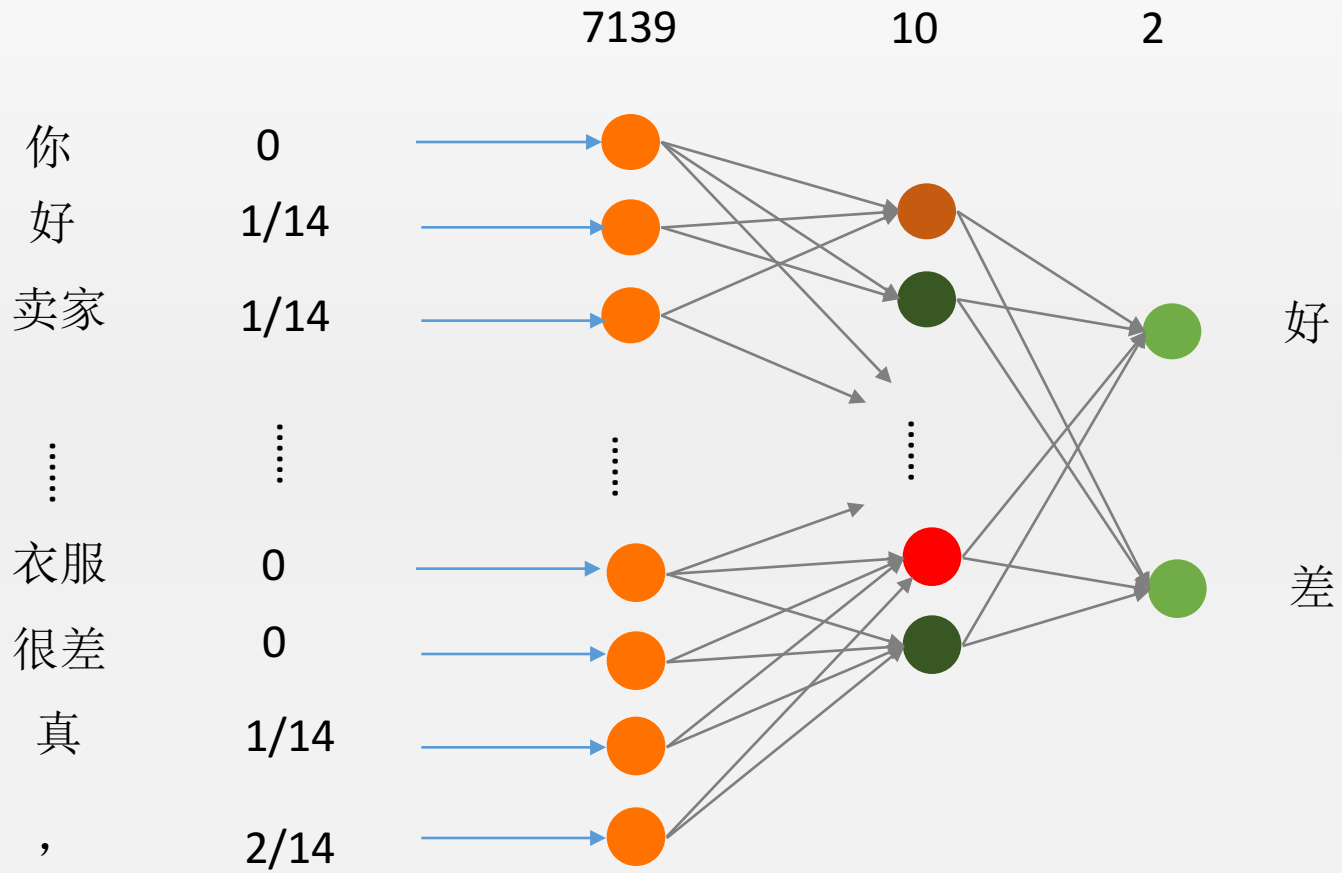
从最后一层开始



第一层



隐含层输入单元的激活



几种判断错误的情况

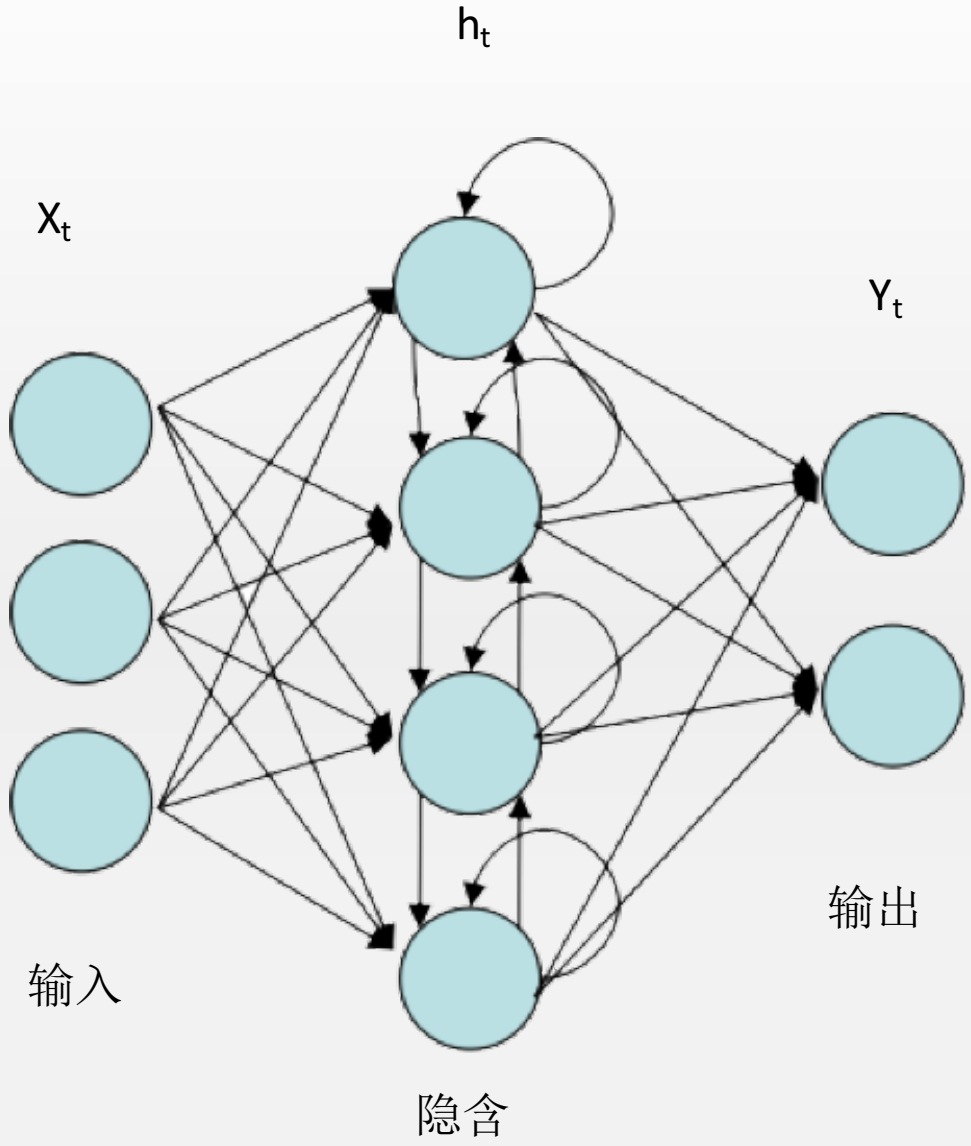
- 数据标注错误:
  - 一分钱，一分货
  - 穿着也舒服，大小合身，质量挺好
- 稀少词，无意义的词
  - asdasdas
- 神经网络有部分没有训练好
  - 不合适穿，给人了　“给”、“人”激活过高，“不”激活低
  - 物流太不给力了　“了”权重过大导致0个神经元激活
- 没有考虑前后相关性
  - 面料不是很好，样式还可以
  - 面料、不是、很、好、样式、还、可以　“好”、“可以”分数高

RNN

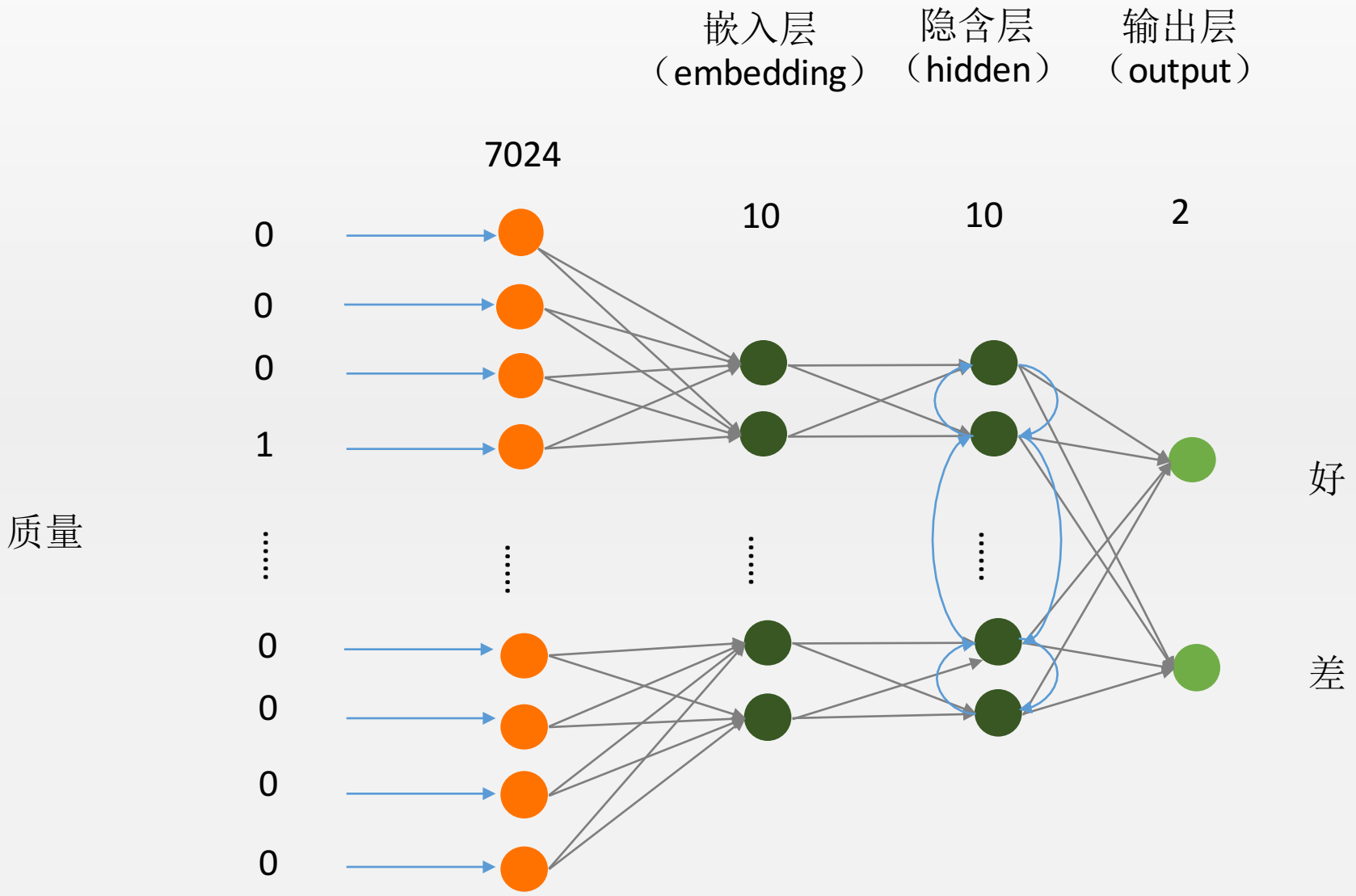
- 隐含层内部存在着连接
- 每一个隐含层单元都与所有其他隐含层单元相连接

$$h_t = \sigma(W_{Xh}X_t + W_{hh}h_{t-1})$$

$$Y_t = \sigma(W_{hY}h_t)$$



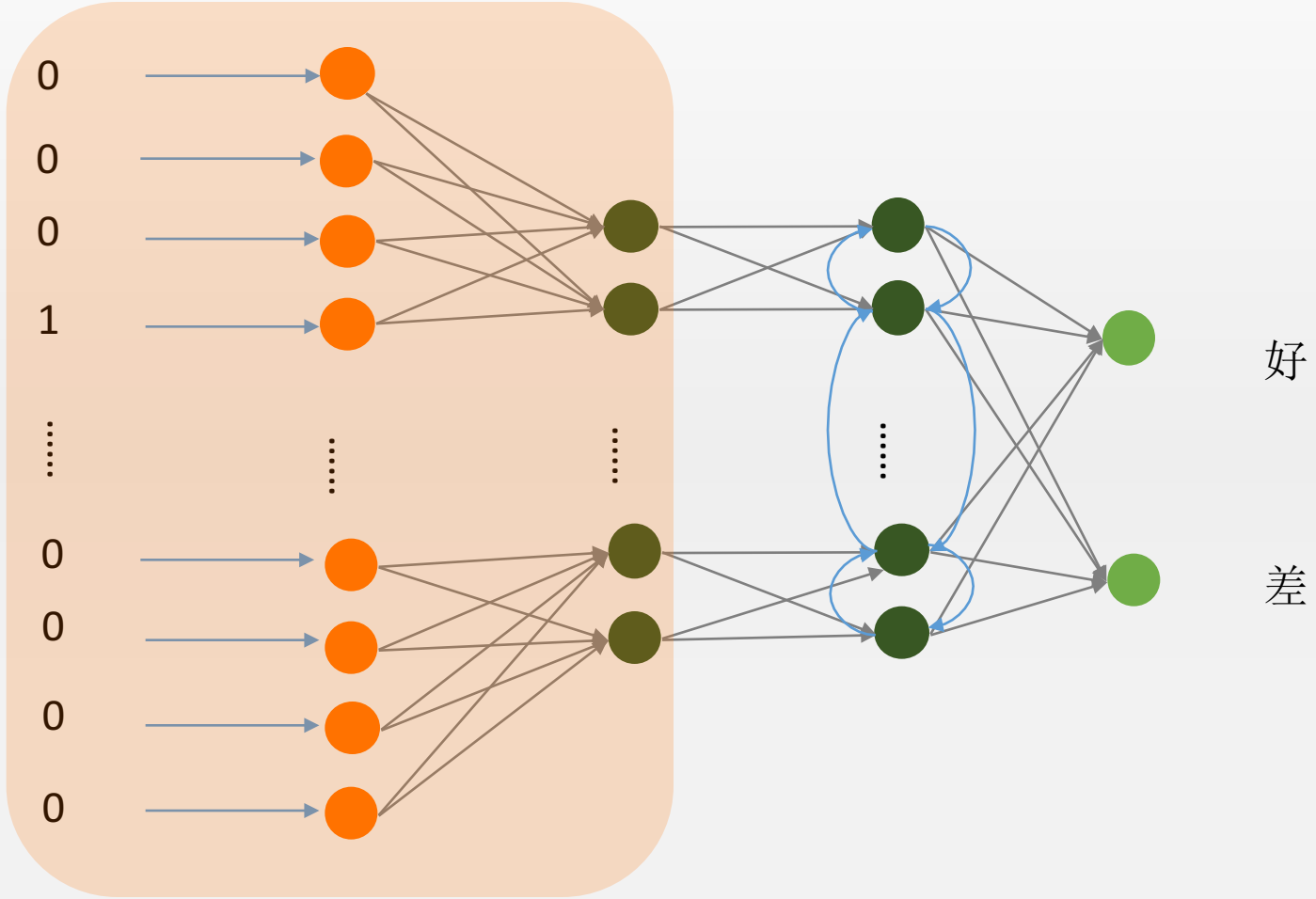
设计一个RNN网络


嵌入层
（embedding）
隐含层
（hidden）
输出层
（output）
7024
10
10
2
0
0
0
1
0
0
0
0
质量
好
差

运行

嵌入层：将单词编号映射为词向量

质量不是很好 4 ⟶

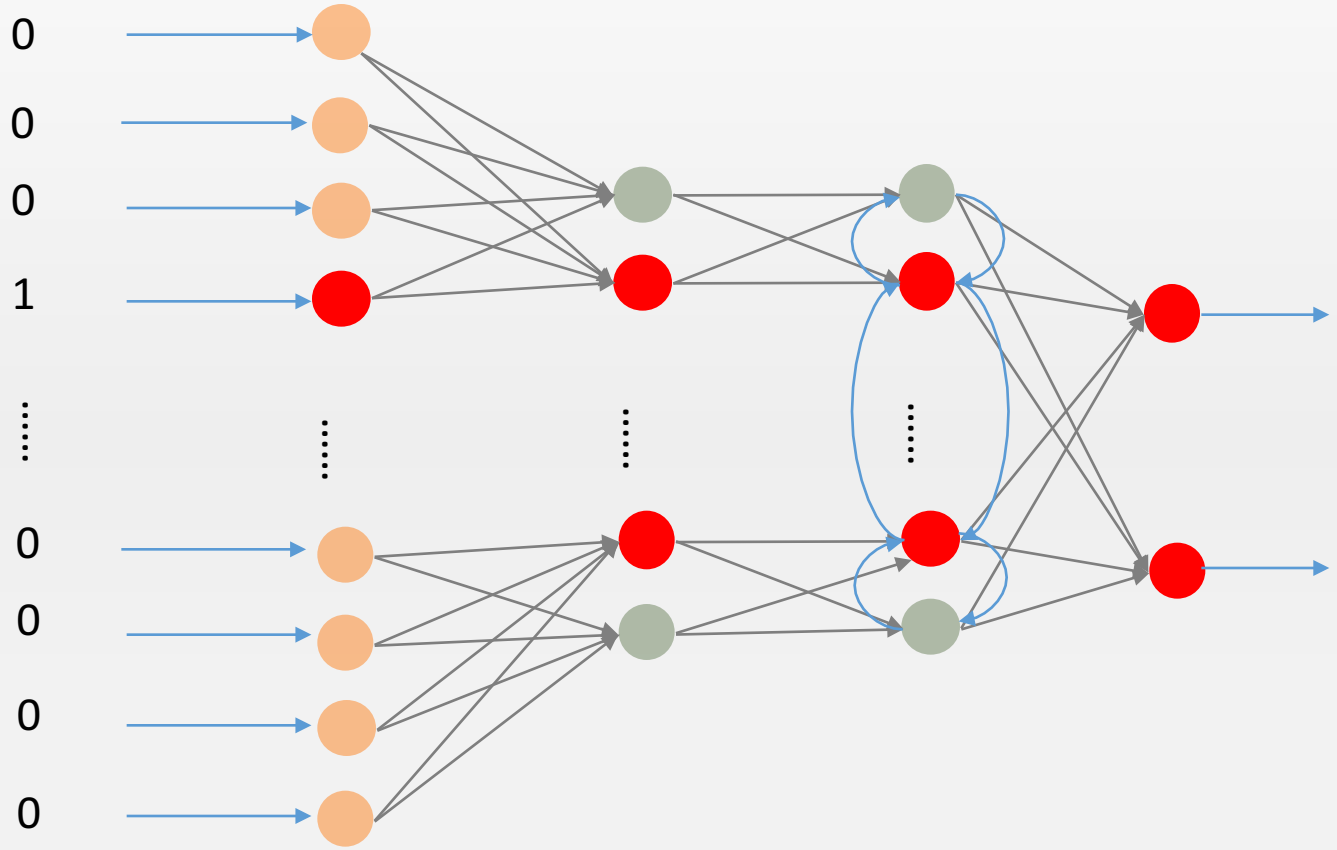


运行1

质量不是很好

$$h_t = \sigma(W_{Xh}X_t + W_{hh}h_{t-1})$$

$$Y_t = \sigma(W_{hY}h_t)$$

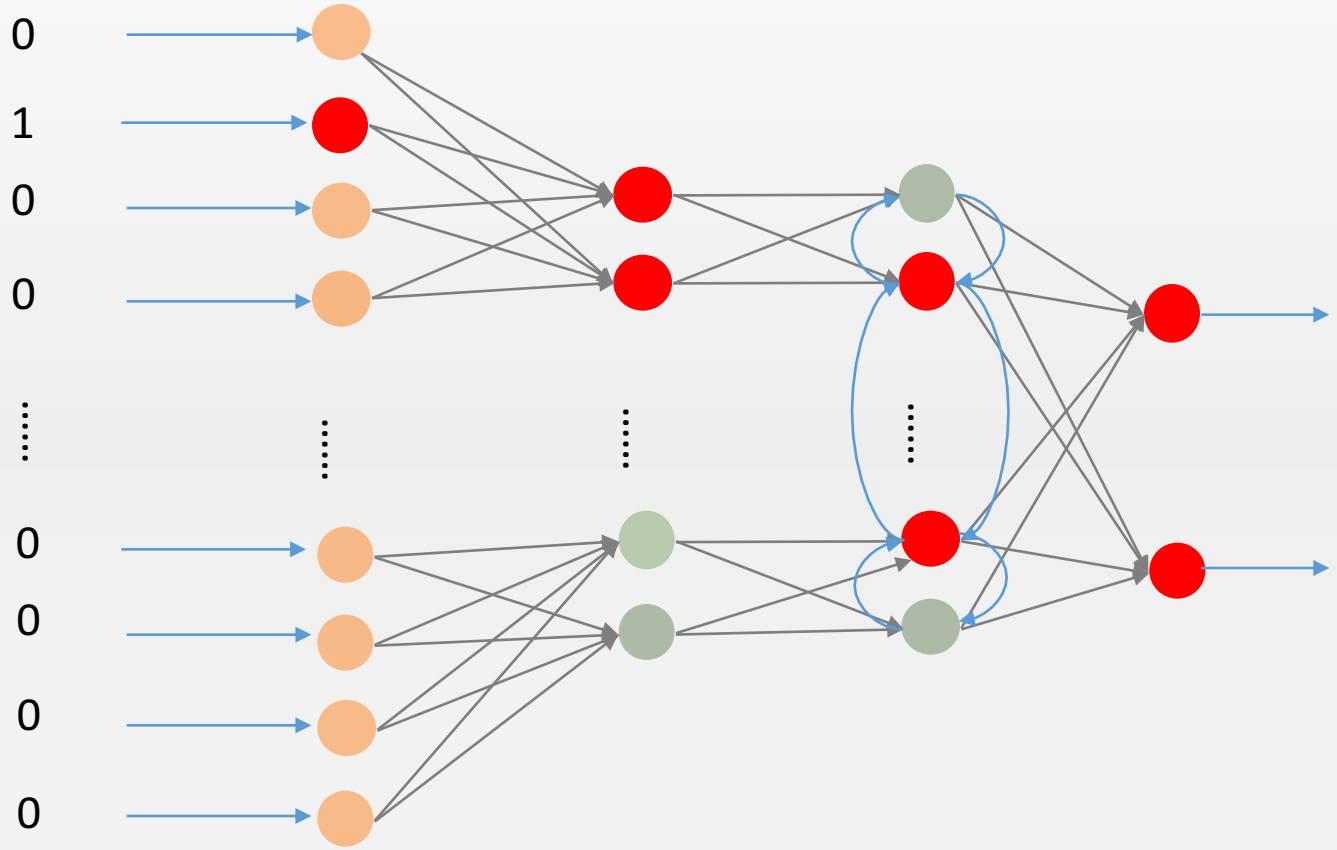


运行2

质量不是很好

$$h_t = \sigma(W_{Xh}X_t + W_{hh}h_{t-1})$$

$$Y_t = \sigma(W_{hY}h_t)$$

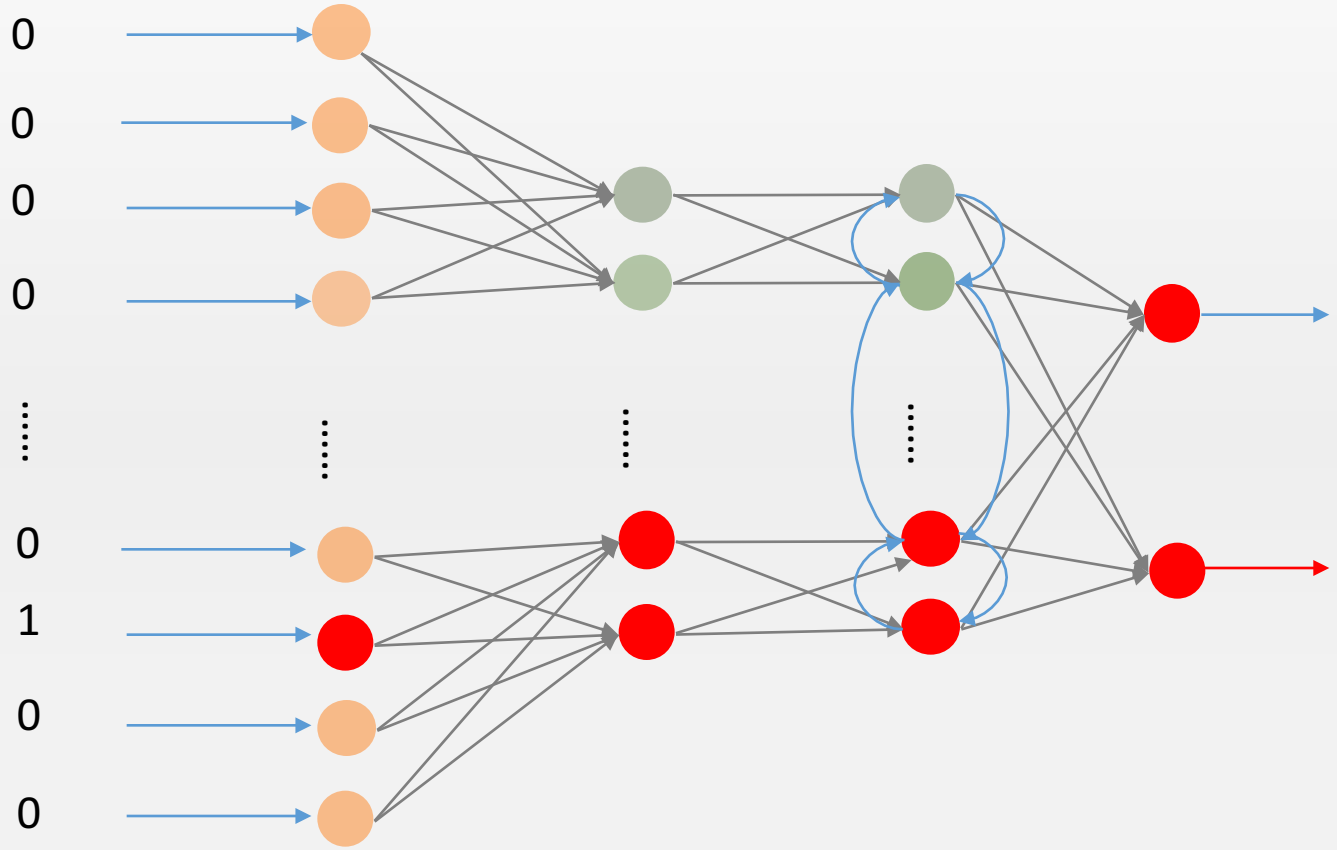


运行3

质量不是很好

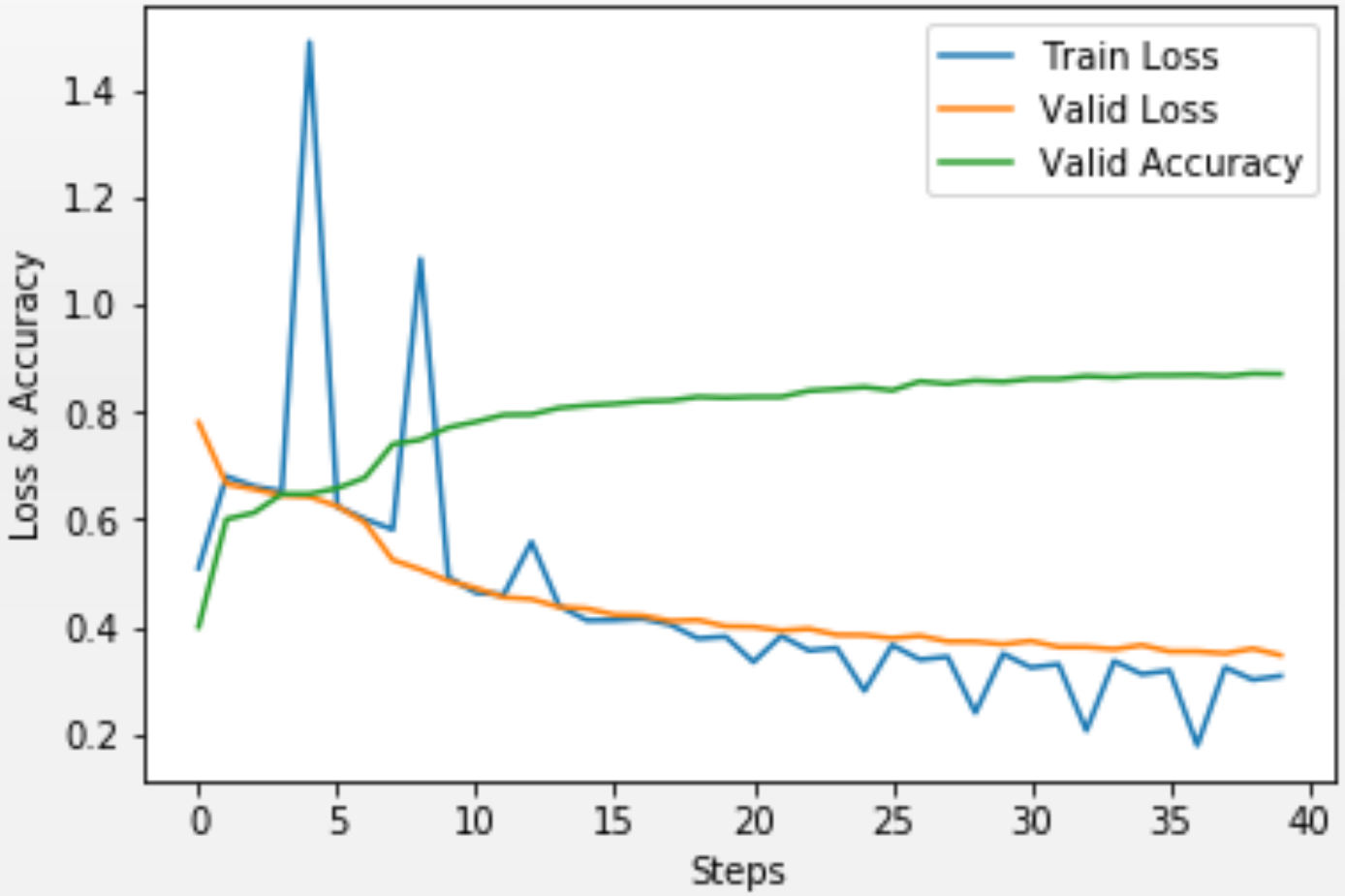
$$h_t = \sigma(W_{Xh}X_t + W_{hh}h_{t-1})$$

$$Y_t = \sigma(W_{hY}h_t)$$



只有最后一个时刻的输出才有效

测试结果



测试数据准确度：0.886

RNN的缺点

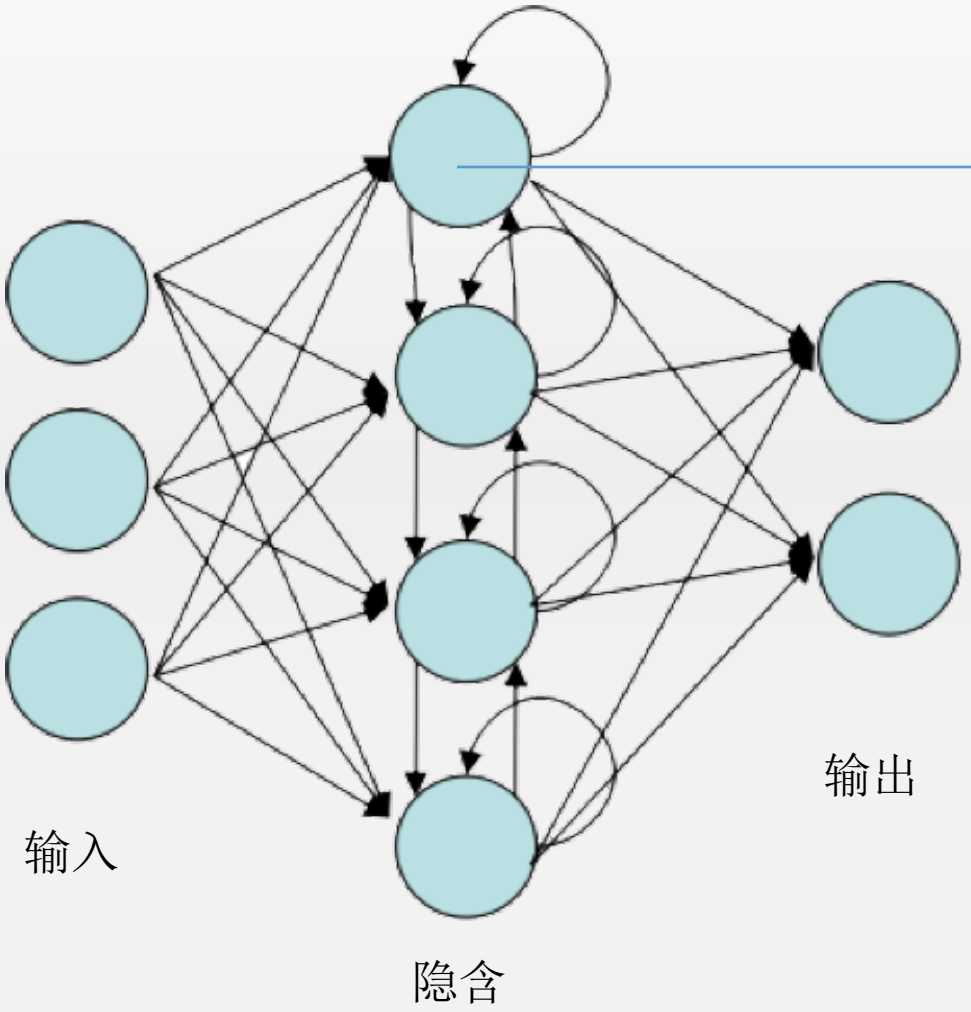
- 运算速度缓慢
- 要小心设置学习率，否则容易出现NAN
- 分类准确度并没有提升很高：无法完成长程记忆

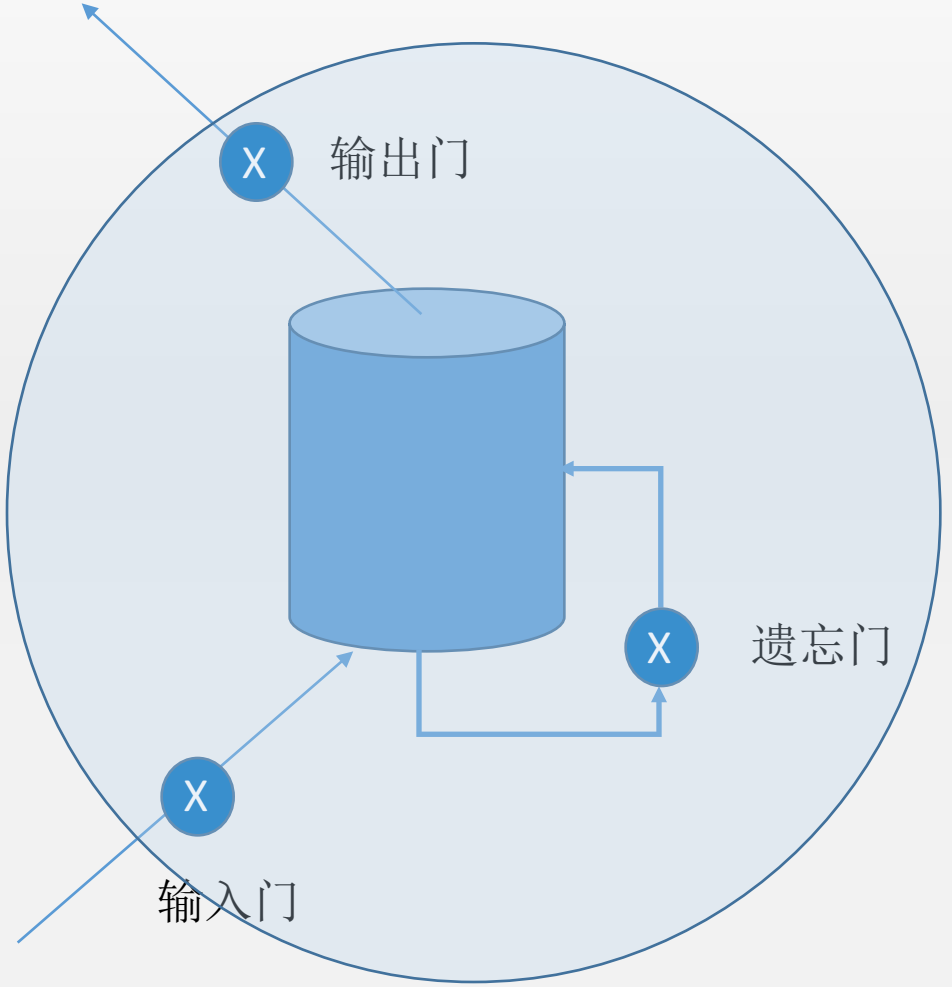
面料不是很好，样式还可以　　√

并非所有的衣服都是跟展示的样品好看！　×

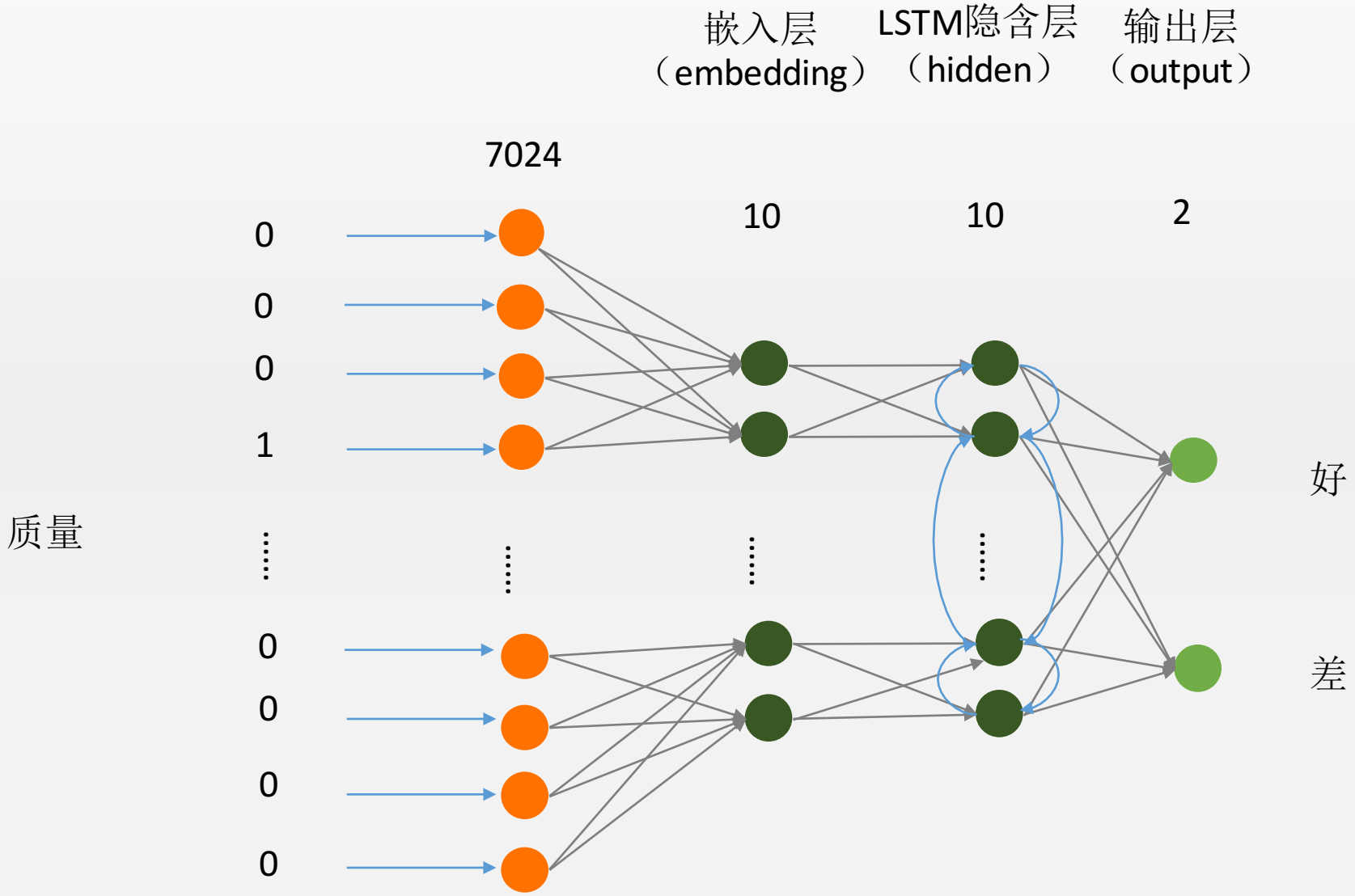
RNN的改进：LSTM（Long Short Term Memory）

LSTM在每一个单元中增加了三个门，以及一个内部状态



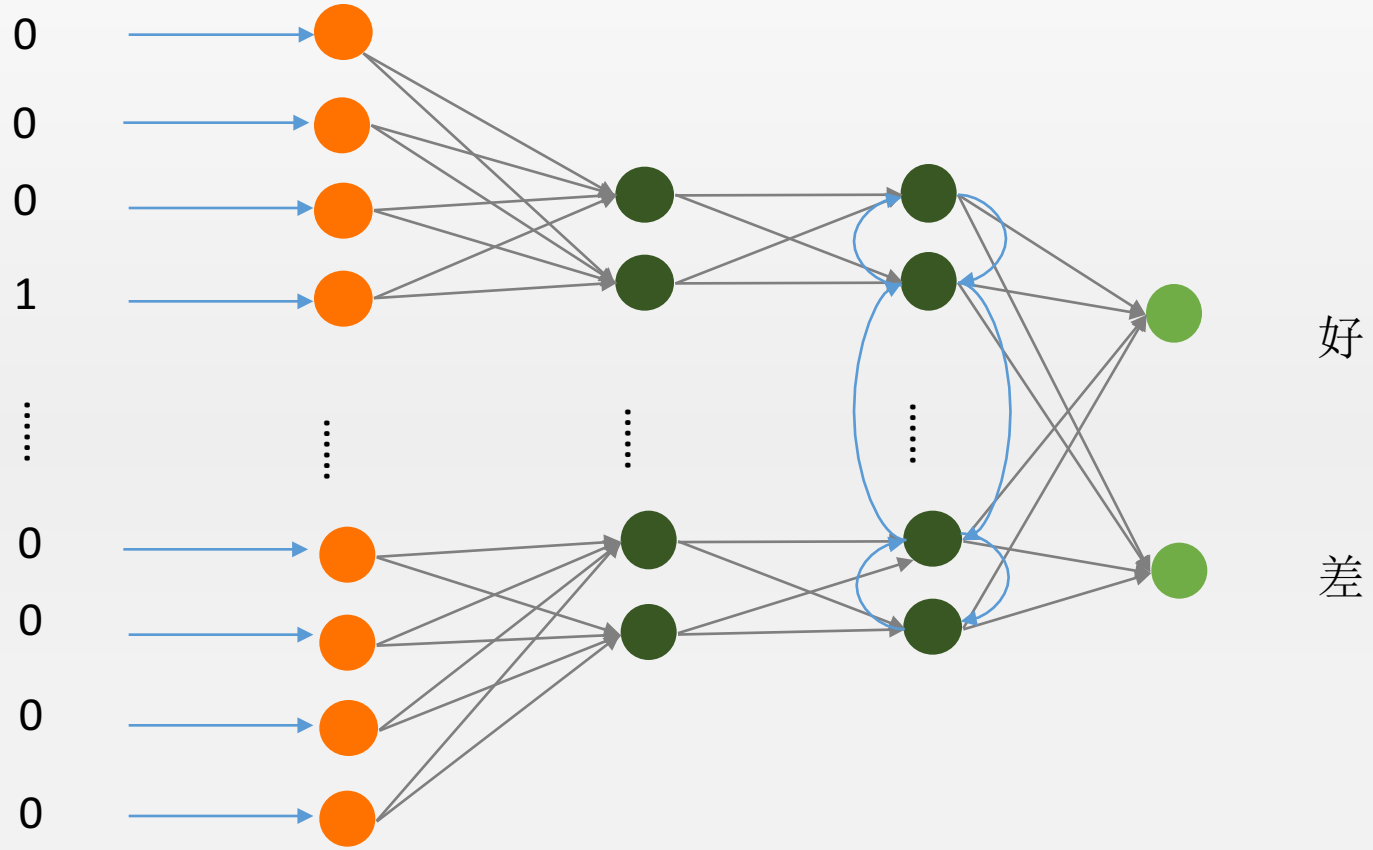


运用PyTorch实现LSTM



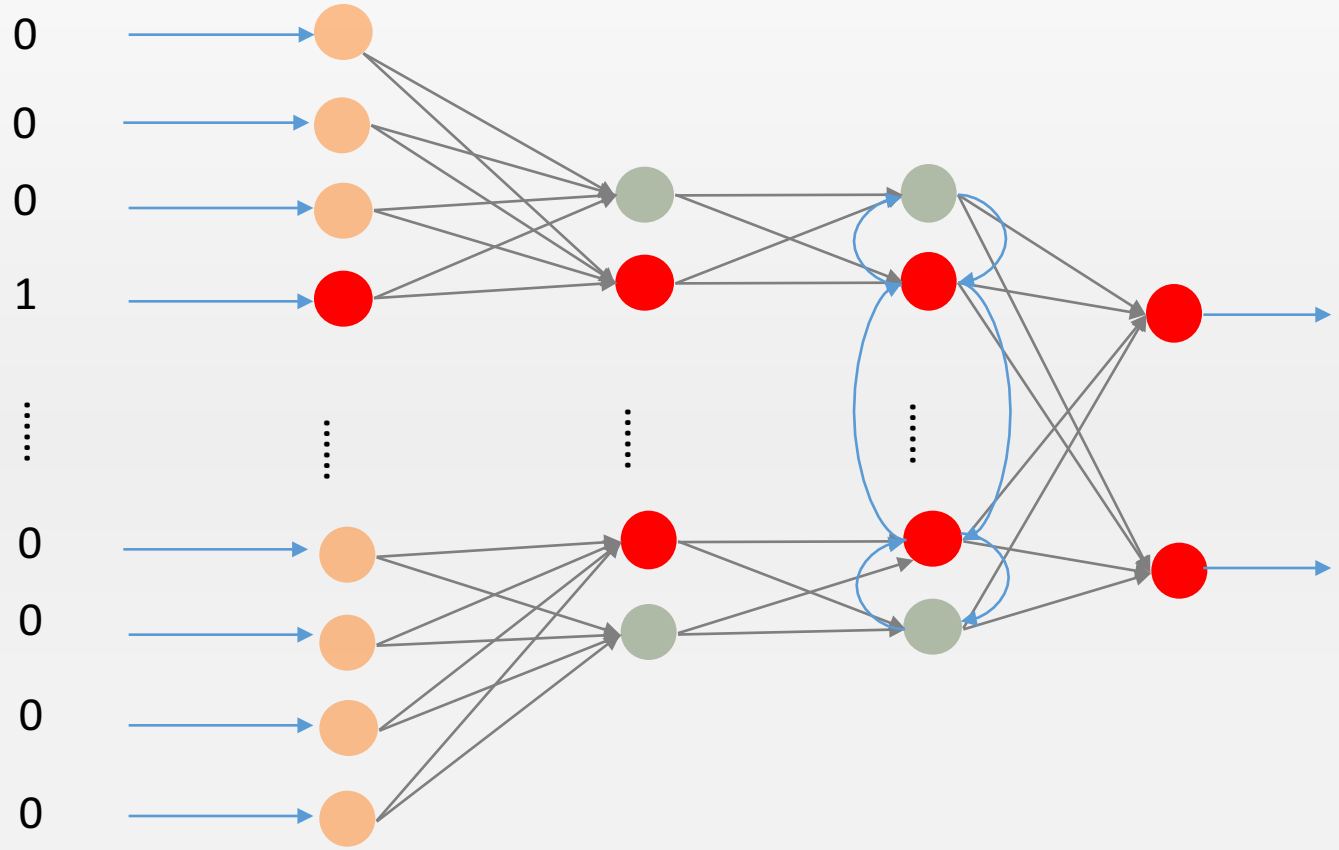
运行

质量不是很好



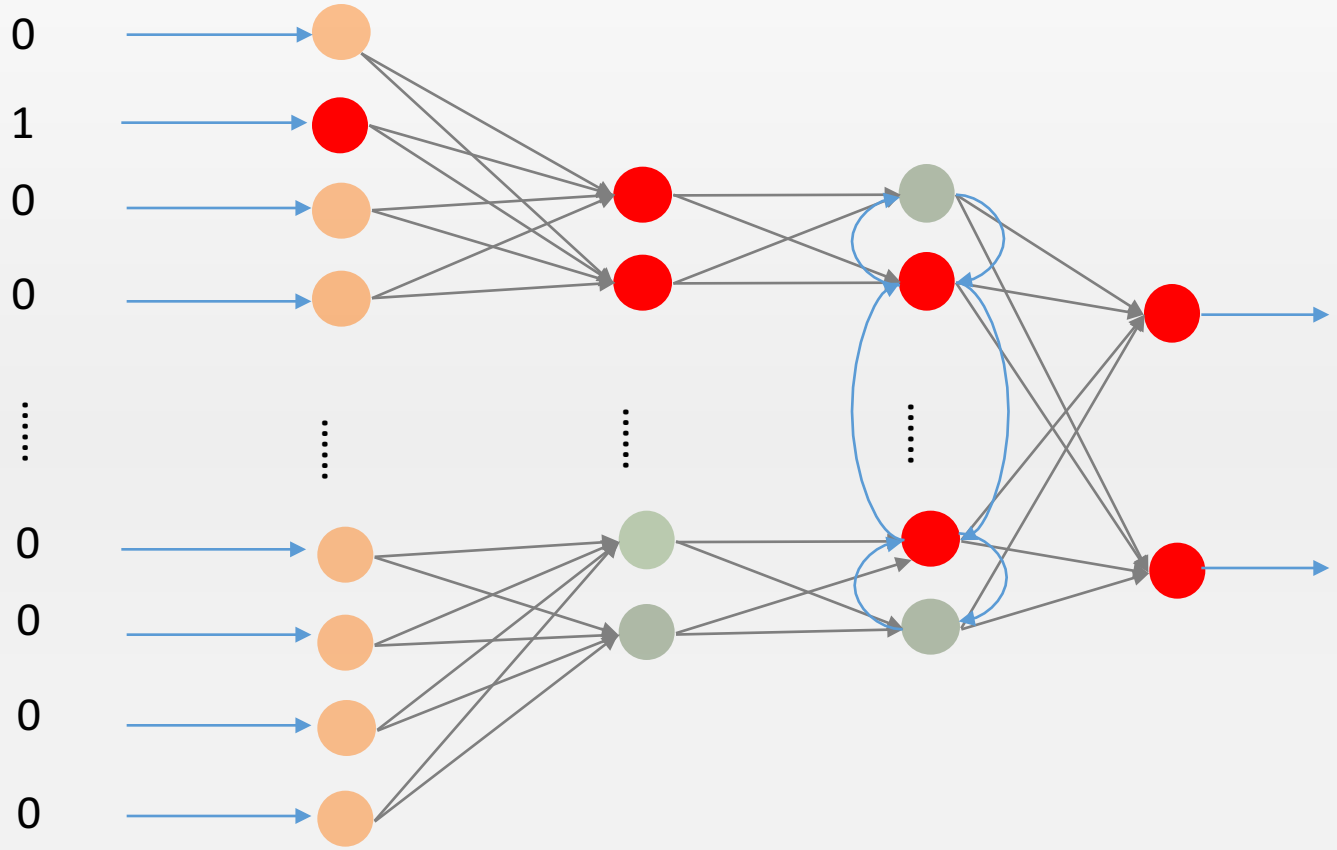
运行1

质量不是很好



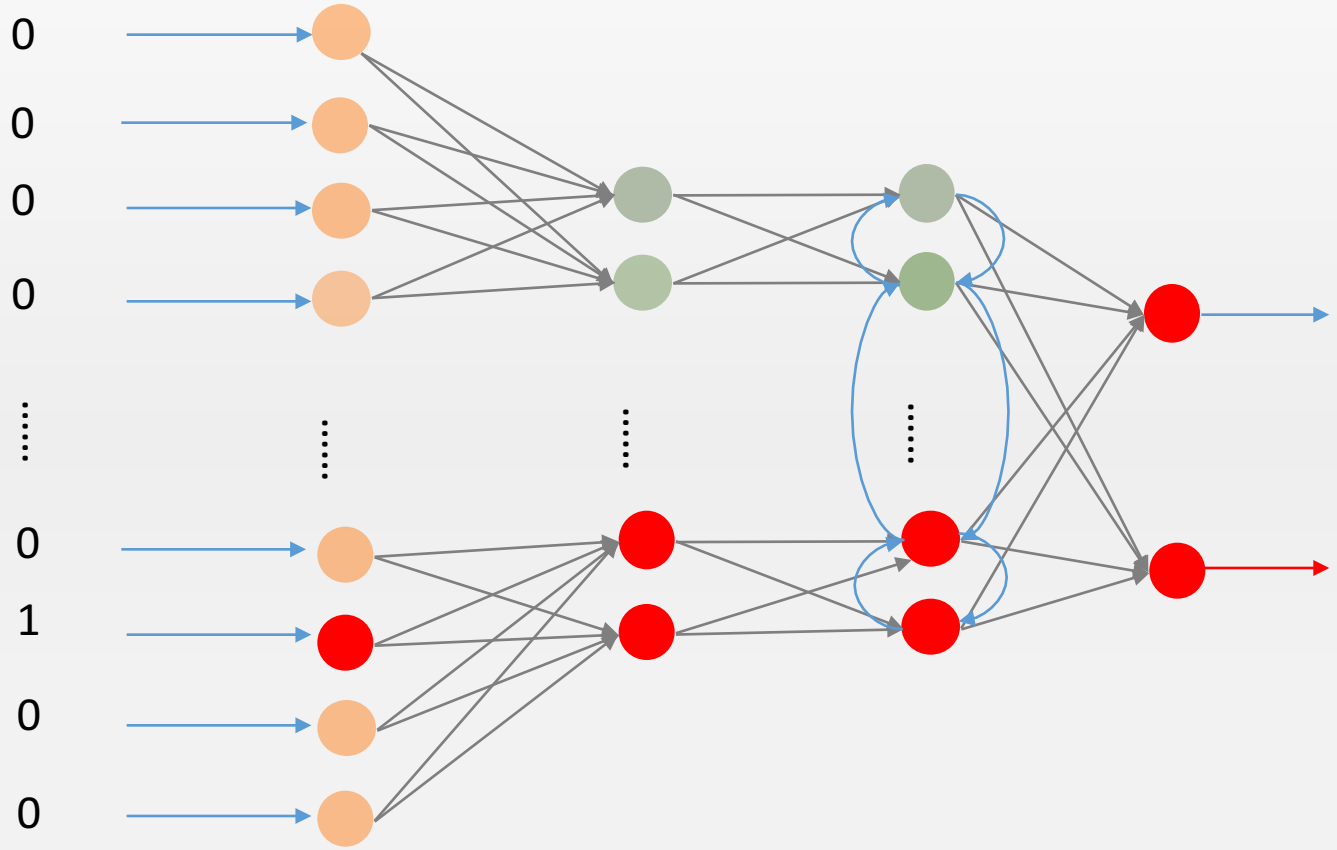
运行2

质量不是很好



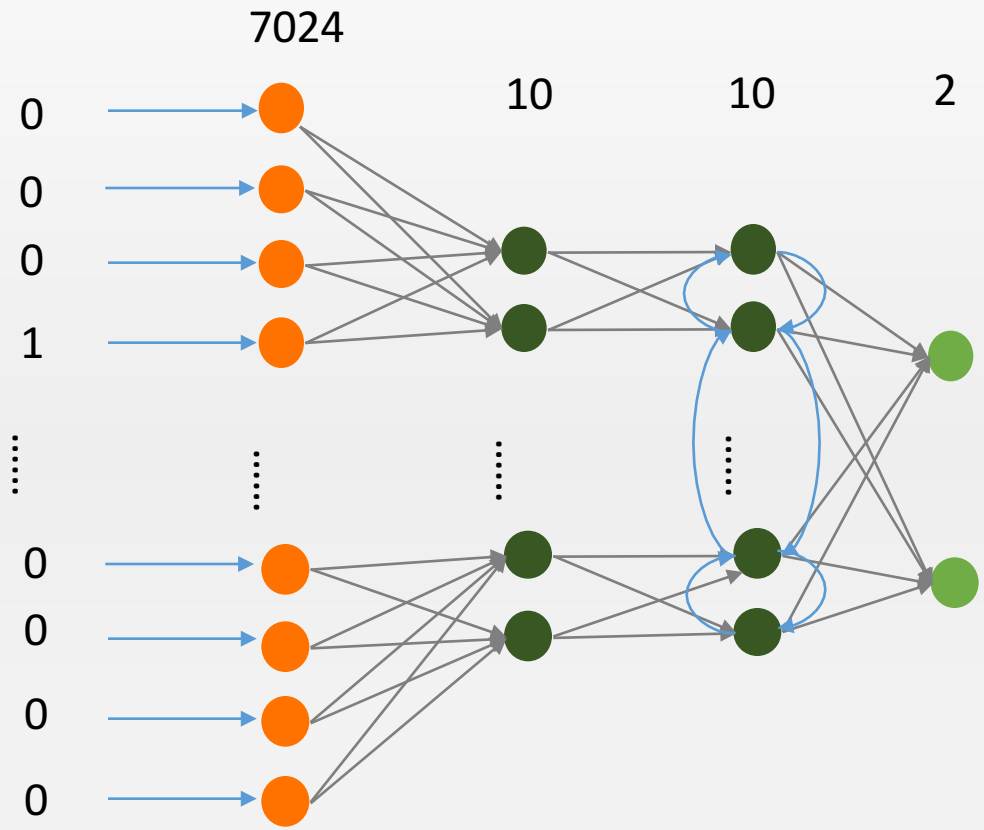
运行3

质量不是很好



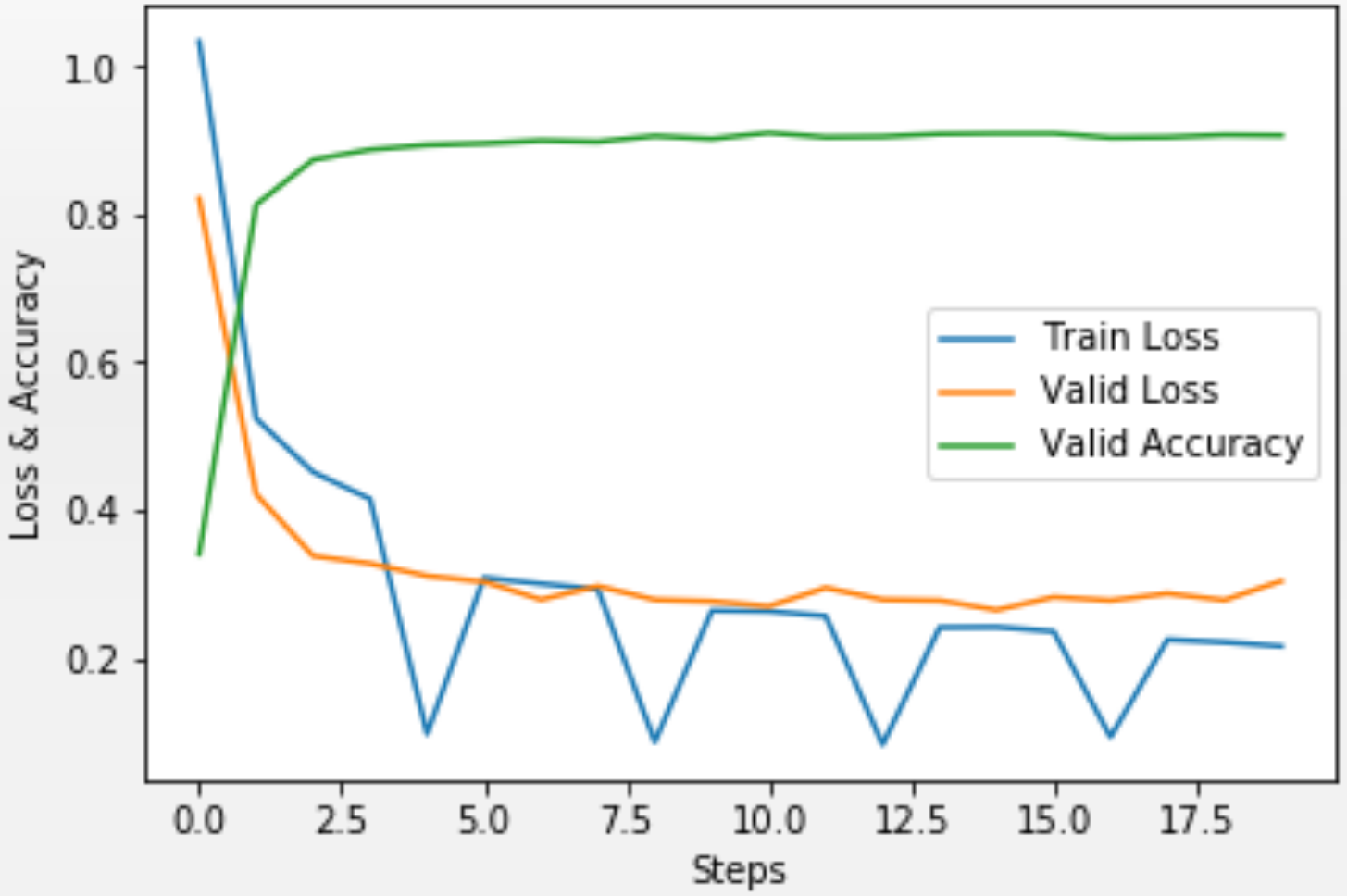
我们仅保留最后一步的输出

LSTM的PyTorch代码




```python
class LSTMNetwork(nn.Module):
    def __init__(self, input_size, hidden_size, n_layers=1):
        super(LSTMNetwork, self).__init__()
        self.n_layers = n_layers
        self.hidden_size = hidden_size

        # LSTM的构造如下：一个embedding层，将输入的任意一个单词映射为一个向量
        # 一个LSTM隐含层，共有hidden_size个LSTM神经元
        # 一个全链接层，外接一个softmax输出
        self.embedding = nn.Embedding(input_size, hidden_size)
        self.lstm = nn.LSTM(hidden_size, hidden_size, n_layers)
        self.fc = nn.Linear(hidden_size, 2)
        self.logsoftmax = nn.LogSoftmax()

    def forward(self, input, hidden=None):
        #词向量嵌入
        embedded = self.embedding(input)

        #PyTorch设计的LSTM层有一个特别别扭的地方是，输入张量的第一个维度需要是时间步，
        #第二个维度才是batch_size，所以需要对embedded变形
        embedded = embedded.view(input.data.size()[0], 1, self.hidden_size)

        #调用PyTorch自带的LSTM层函数，注意有两个输入，一个是输入层的输入，另一个是隐含层自身的输入
        # 输出output是所有步的隐含神经元的输出结果，hidden是隐含层在最后一个时间步的状态。
        # 注意hidden是一个tuple，包含了最后时间步的隐含层神经元的输出，以及每一个隐含层神经元的cell的状态

        output, hidden = self.lstm(embedded, hidden)

        #我们要把最后一个时间步的隐含神经元输出结果拿出来，送给全连接层
        output = output[-1,...]

        #全链接层
        out = self.fc(output)
        # softmax
        out = self.logsoftmax(out)
        return out

    def initHidden(self):
        # 对隐单元的初始化
        # 对引单元输出的初始化，全0.
        # 注意hidden和cell的维度都是layers,batch_size,hidden_size
```

LSTM的效果



测试数据准确率：0.918

要点重述

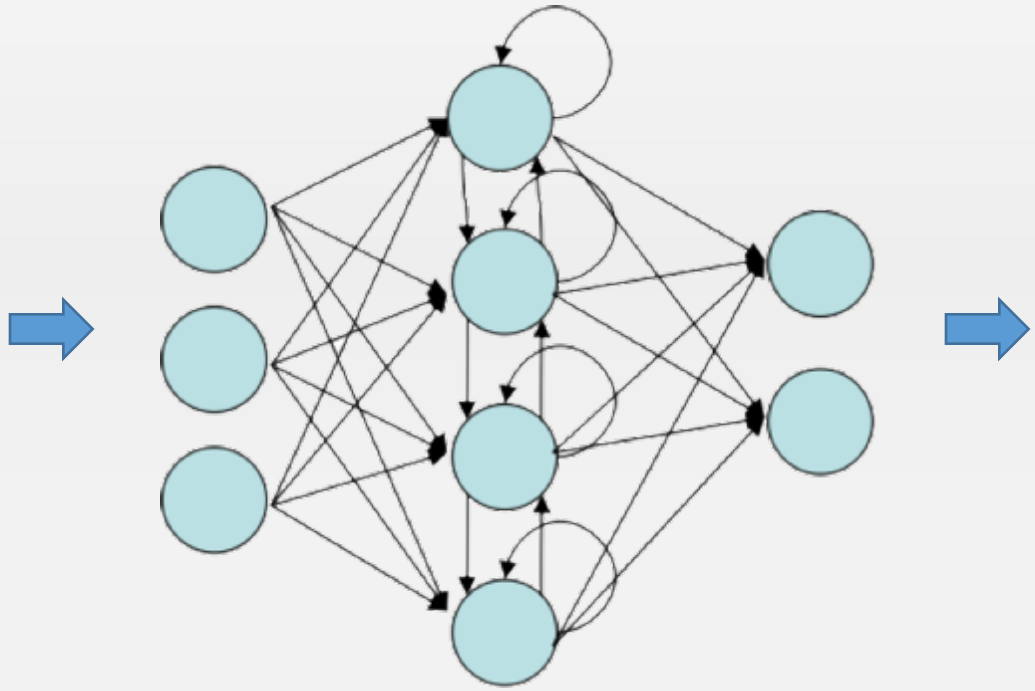
- 文本分类、情绪分类
- 词带模型
- RNN模型
- LSTM模型

作业：名称国籍识别器

- 请设计一个RNN网络，要求输入一个英文单词（有可能是拼音），输出这个名称对应的国别

Zhang Jiang

Jake



中国

美国

如何结合标签信息？

敬请期待

- 张江

集智AI学园
campus.swarma.org