模157. 2020.12.12.

·回顾：

线性回归 —Sigmoid ($\frac{1}{1+e^{-z}}$)→ LR 逻辑回归；二分类.

多分类，二分类 回归 时序

ID3/C4.5
树 → CART —类别型特征→ 基尼 GINI，找最优分裂节点

{bagging / boosting} ⇒ RF → GBDT ⇒ XGB → LGB （随机森林）

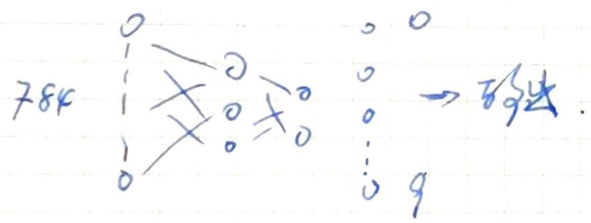
时序 —参数→ {模型 / Arima}

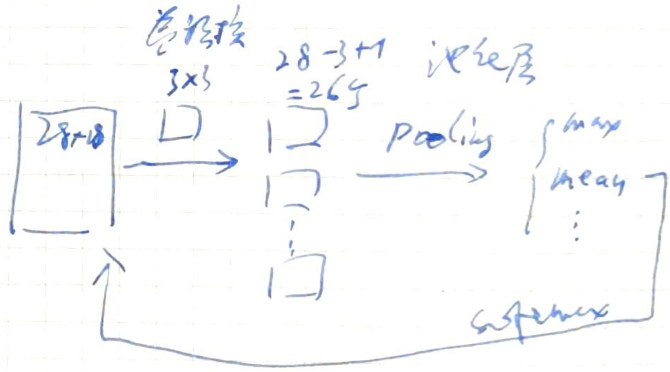
BP —反向传播→ LSTM (RNN) → {图片 / 文本 / Embedding}

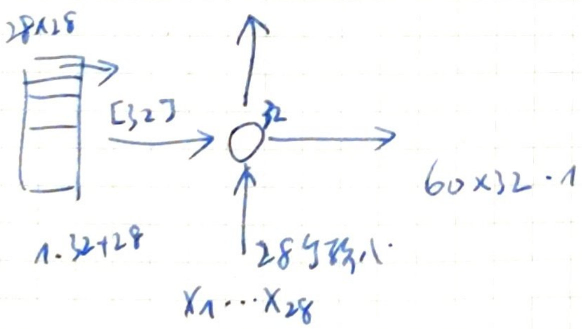
梯度爆炸、消失问题. 时序的. bert ?

推荐系统：User × Item —组合→ GBDT + LR

·作业：MNIST. 28x28 = 784

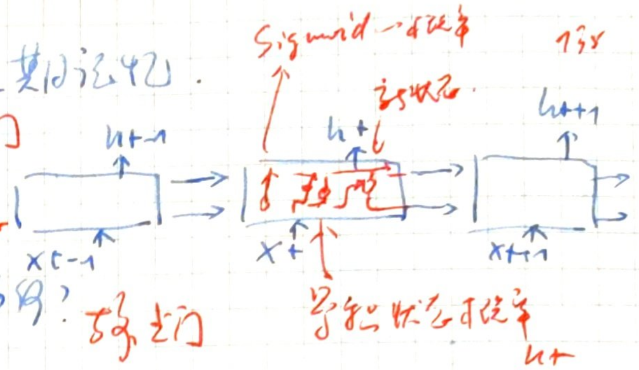


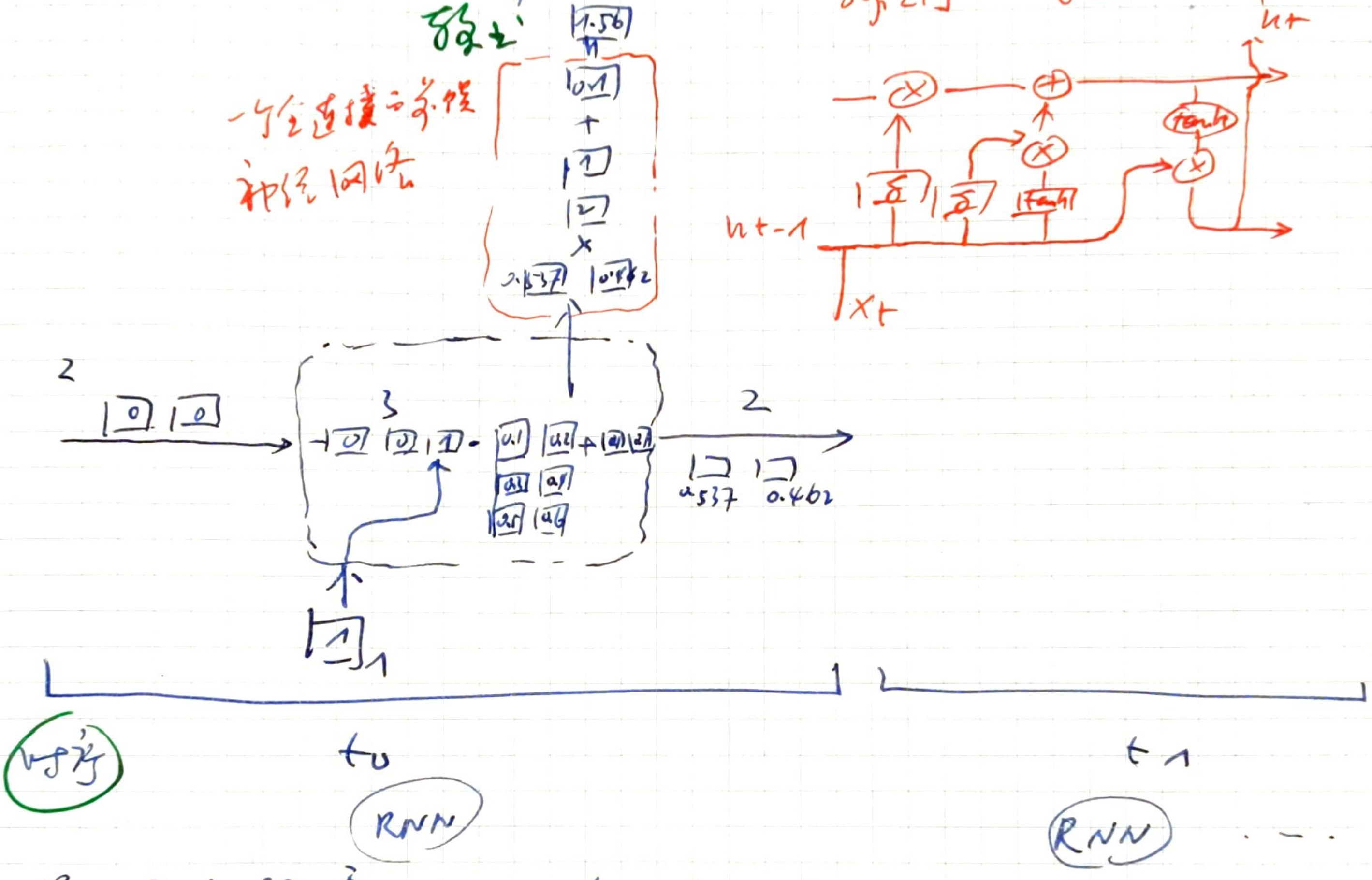




LSTM: Long short Term 长短期记忆.

- 应忘掉哪些细胞状态. 遗忘门
- 应加入哪些新的状态. 细胞状态
- 根据当前状态和输入, 输出是什么? 输出门





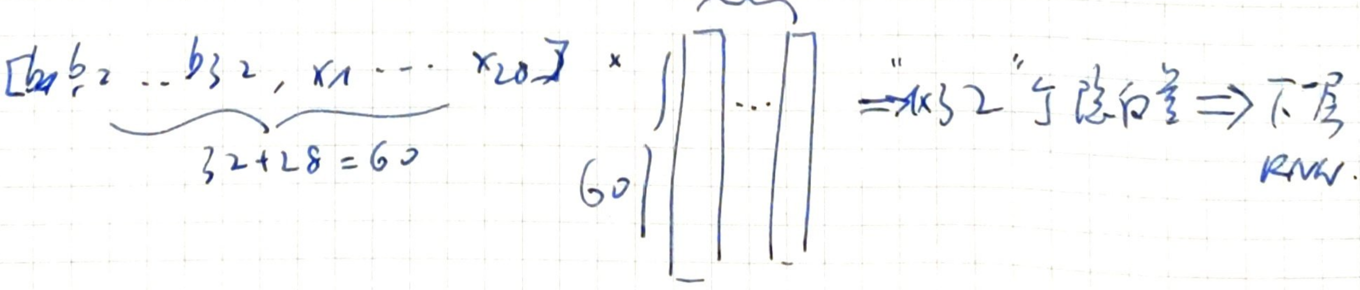
RNN 正向传播: $t_0 \to t_1 \to t_2 \dots t_n$.

I am a 清华 student.

t_0 t_1 t_2

MNIST, 手写识别: 28×28.

LSTM 层32个节点. => 32个隐变量.



Count Vektorizer：文本特征提取方法.

- 只考虑每种词汇在该训练文本中出现的频率.

每段文 → [—— ——] — n-words.

另段文本 → [—— ——]

TF-IDF 原理：一个词在当前文章中出现多，在所有文档中出现得少，表示能代表当篇文章.

词频：$TF_x = \frac{\text{词x出现的次数}}{\text{所有的词条数量}}$ 即 $\frac{\text{词数}}{\text{文章总词数}}$

问题：① 长文本可能出现的次数偏多.
② 通用词语词数较大，但不表示主题，反而某些小词数的代表文章

文权重：预测主题强 ↑ 权重大；弱 ↓ 小

所有的统计文章中，一些词只在很少几篇文章中出现，则其主题相关. ↲ IDF

Term Frequency - Inverse Document Frequency

逆文件频率.

$IDF = \log\left(\frac{\text{语料库文章总数}}{\text{包含词条w的文章总数}+1}\right)$

TF - IDF = TF · IDF ← 权重，到处都存在的词，权重小.

↑ 在文章中重要性   逆文件频率

Tokenizer & pad-sequence 原理

↑ 分类，统计词数.   ↑ 填充后，截断后（或删）都相同的词序列.（keras 只接受等长输入）

~~补充~~ 截断…

from sklearn.feature_extraction.text import CountVektorizer, TFidfVektorizer.

① 先 Count 统计 CountVektorizer().fit_transform(text) TfidfTransformer

② vetori.get_feature_names() / vetori.vocabulary_ TF-IDF

③ count.toarray

[——] ← 文章1
[- ——] 文章2
⋮
[- -] 文章n.

transformer = TfidfTransformer()
tfidf_matrix = transformer.fit_transform(count)

Tokenizer: from keras.preprocessing.text import Tokenizer

tok = Tokenizer().fit_~~transform~~ on_text (text)

tok.word_index / tok.word_counts

哪些词出现

tok.texts_to_matrix(text) → [0..1,1,..011..0]
[ ]
:
[ ]

tok.texts_to_sequences(text)

句子中词汇编号(顺序)

↳ [[34,122], [..], [–]]

Pad-sequences: keras.preprocessing.sequence import pad_sequences

pad_sequences(text, maxlen=100)

考虑到!! 思路: 先用TF-IDF / Tokenizer / pad_sequences.

Lgb ← numpy sparse (from scipy)

定义信息: 计算出句子的字符向量, 然后再喂给其它机器学习模型.

CountVectorizer.

通过词频及相互组合的关联特性 ⇒ 是否定义.

推荐系统、文本 (哪些字词?) 多词组合 ⇒ 是否感兴趣.

推荐系统：

- 用户：解决"信息过载"，高效获取感兴趣的信息.
- 企业：吸引、留存用户，增加粘性，提高用户转化率.
  - 视频网站：观看时长 (VR)
  - 电商：购买转化率.
  - 新闻：点击率 CTR.

— 热门推荐
— 个性化推荐
— [illegible]推荐
— 搜索推荐

- base：人 (User) / 物 (Item) / 场景 (Context) → 时间、地点、用户状态.
  历史行为 / 人口属性 / 关系网络.
- 效果：相关性 / 新颖性 / 惊喜性 / 多样性
- 目标：
  - 用户满意度，以及企业业务指标.
  - 模型指标：RMSE / 召回率 / 精准率
  - top N：召回和排序.
  - 覆盖率：挖掘长尾，多样性
- 流程
  - 召回 → 排序 → 离线 + 在线模型 → 实时针对用户推荐

  100w Item —挑→ 100 → Embedding + 时序特征 → 点击

- 难点：
  - 数据稀疏的问题.
  - 用户兴趣转变.
  - 实时更新.
  - 冷启动问题.

显式反馈：明确喜好.
隐式反馈：不明确.

ML/DL　NLP　CV　多模态

多种知识结合 获得的 Embedding

← UserCF / ItemCF 见另笔记