Lesson 4
STRUCTURED, TIME SERIES, & LANGUAGE MODELS

2018年7月

上海交通大學
SHANGHAI JIAO TONG UNIVERSITY

目录 Contents

1 博文推荐

2 Overfitting与Dropout

3 结构化时间序列模型介绍

4 自然语言处理介绍

目录 Contents

1 博文推荐

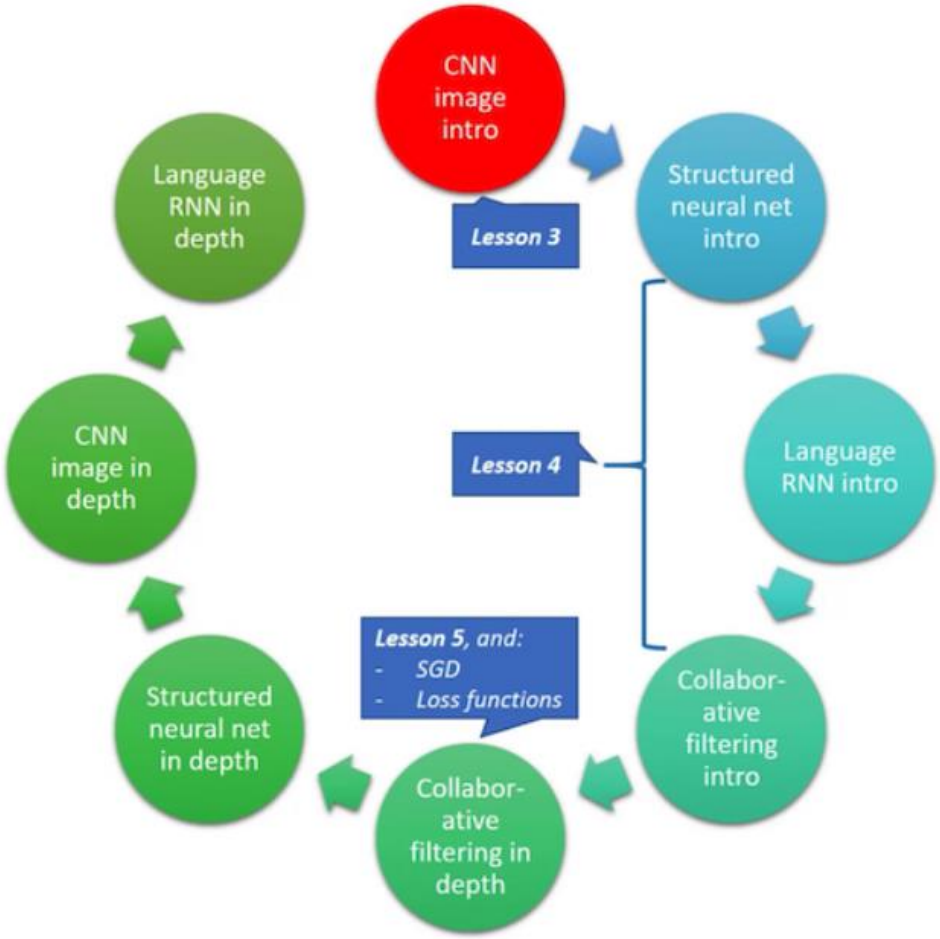
2 Overfitting与Dropout

3 结构化时间序列模型介绍

4 自然语言处理介绍

博文推荐

- Improving the way we work with learning rate
- The Cyclical Learning Rate technique
- Exploring Stochastic Gradient Descent with Restarts (SGDR)
- Transfer Learning using differential learning rates
- Getting Computers To See Better Than Humans



目录 Contents

1 博文推荐

2 Overfitting与Dropout

3 结构化时间序列模型介绍

4 自然语言处理介绍

Overfitting——过拟合

- 原因：

  1. 训练数据较少，不够泛化

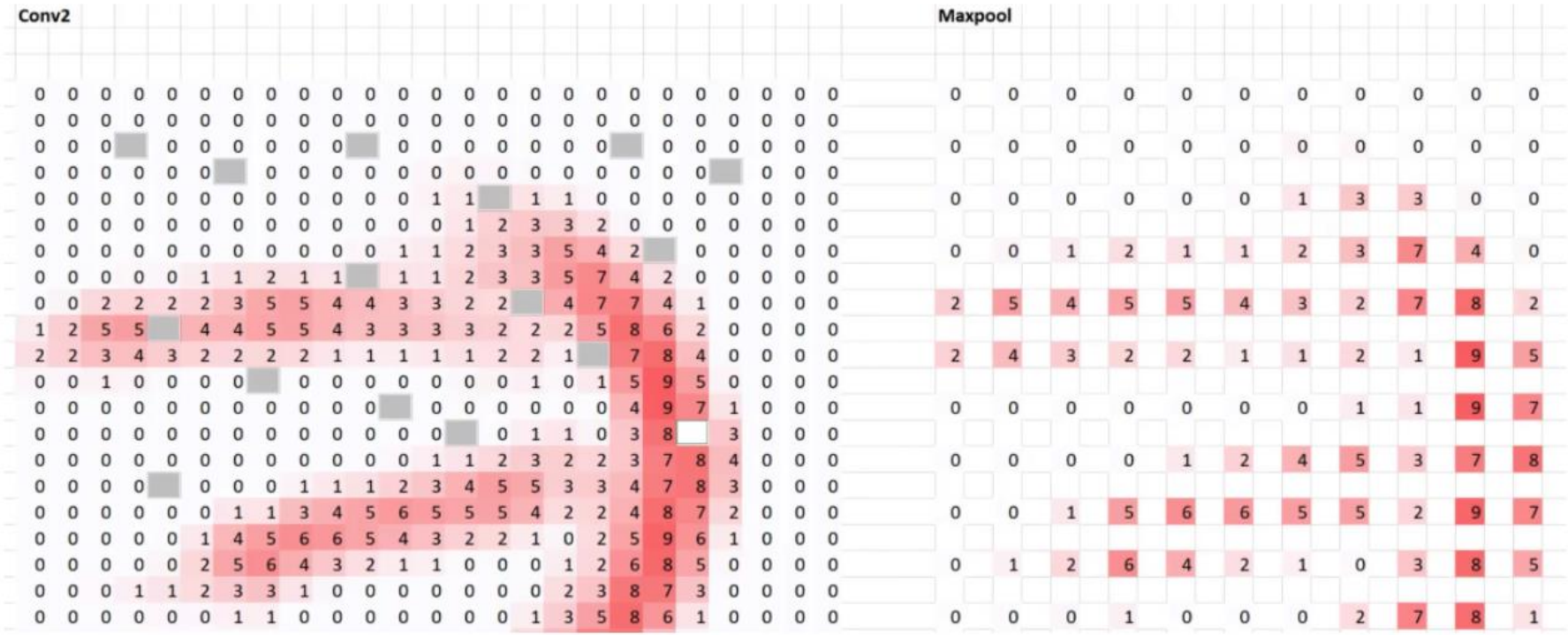
  2. 模型太过复杂，参数过多

- 表现：training loss << validation loss,
- 解决方案：dropout等

本部分示例代码为第一课的dogbreed

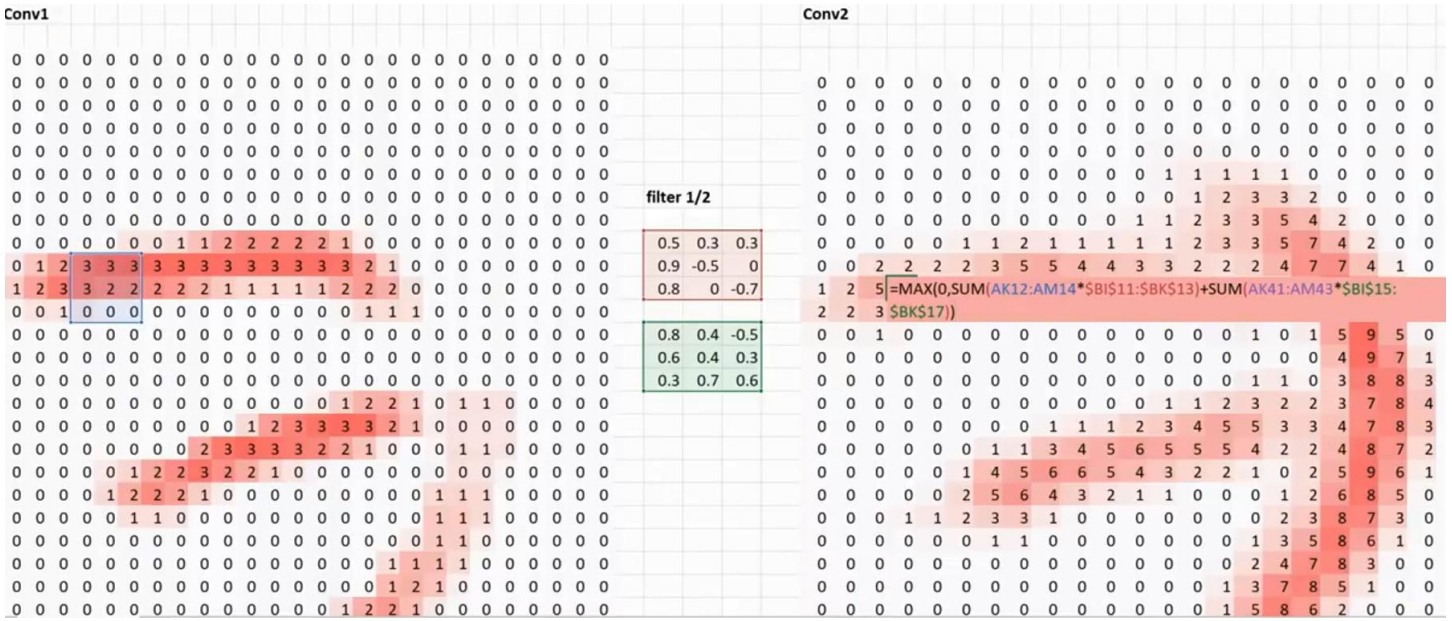
Dropout

- 原理：每次迭代更新时按照一定概率随机丢弃一部分神经元，相当于训练多个不同结构的网络，可以理解为一种Ensemble方法
- 例子：池化层前以p=0.5的概率随机删除activations



Dropout

- 另一种情形：在两个卷积层之间进行dropout
- 由于dropout会使其均值降低，使得卷积输出的值降低，之后的每一层也相应降低，而测试时不采用dropout，因此可以在训练时将参数乘以1/p后再进行dropout，使均值不变
- 或者也可以在测试时将参数乘以p，以达到和训练模型的一致



Dropout

- Tips：根据实际情形设置各层dropout值

```
learn = ConvLearner.pretrained(arch, data, ps=[0., 0.2],
            precompute=True, xtra_fc=[512]); learn

Sequential(
  (0): BatchNorm1d(4096, eps=1e-05, momentum=0.1, affine=True)
  (1): Linear(in_features=4096, out_features=512)
  (2): ReLU()
  (3): BatchNorm1d(512, eps=1e-05, momentum=0.1, affine=True)
  (4): Dropout(p=0.2)
  (5): Linear(in_features=512, out_features=120)
  (6): LogSoftmax()
)
```

目录 Contents

1 博文推荐

2 Overfitting与Dropout

3 结构化时间序列模型介绍

4 自然语言处理介绍

数据与变量

数据类型：

- Structure：可以用数据库二维逻辑表示的数据
- Unsturcture：图片、音频、视频等不方便使用数据库二维逻辑表示的数据

变量类型：

- Categorical：分类的类别，有层次的划分，如商店类型
- Continuous：由连续的数字组成，如竞争对手开店的距离，也可表示为非连续变量

```
cat_vars = ['Store', 'DayOfWeek', 'Year', 'Month', 'Day', 'StateHoliday', 'CompetitionMonthsOpen',
    'Promo2Weeks', 'StoreType', 'Assortment', 'PromoInterval', 'CompetitionOpenSinceYear', 'Promo2SinceYear',
    'State', 'Week', 'Events', 'Promo_fw', 'Promo_bw', 'StateHoliday_fw', 'StateHoliday_bw',
    'SchoolHoliday_fw', 'SchoolHoliday_bw']

contin_vars = ['CompetitionDistance', 'Max_TemperatureC', 'Mean_TemperatureC', 'Min_TemperatureC',
   'Max_Humidity', 'Mean_Humidity', 'Min_Humidity', 'Max_Wind_SpeedKm_h',
   'Mean_Wind_SpeedKm_h', 'CloudCover', 'trend', 'trend_DE',
   'AfterStateHoliday', 'BeforeStateHoliday', 'Promo', 'SchoolHoliday']

n = len(joined); n
```

本部分示例代码为lesson3-rossmann.ipynb

对数据的处理

- 变量单独化、数值化
- do_scale:神经网络希望输入数据的均值为0，标准差为1。这里取数据后减去均值，除以标准差，以获得一个标准化的均值与标准差。
- 缺失值处理：Categorical 赋值为0；Continuous：用中位数替换缺失的值，并创建一个新的布尔列，表示它是否缺失。

```
joined_samp.head(2)
```

Date	Store	DayOfWeek	Year	Month	Day	StateHoliday	CompetitionMonthsOpen	Promo2Weeks	StoreType	Assortment	...	Max_Wind_SpeedKm_h	Mean_\
2014-01-08	781	3	2014	1	8	False	24	0	a	a	...	29.0	
2014-11-22	626	6	2014	11	22	False	12	0	c	c	...	23.0	

2 rows × 39 columns

```
df.head(2)
```

Date	Store	DayOfWeek	Year	Month	Day	StateHoliday	CompetitionMonthsOpen	Promo2Weeks	StoreType	Assortment	...	Mean_Wind_SpeedKm_h	Cloud
2014-01-08	781	3	2	1	8	1	25	1	1	1	...	0.367717	1.4
2014-11-22	626	6	2	11	22	1	13	1	3	3	...	0.367717	-0.3

2 rows × 40 columns

训练步骤

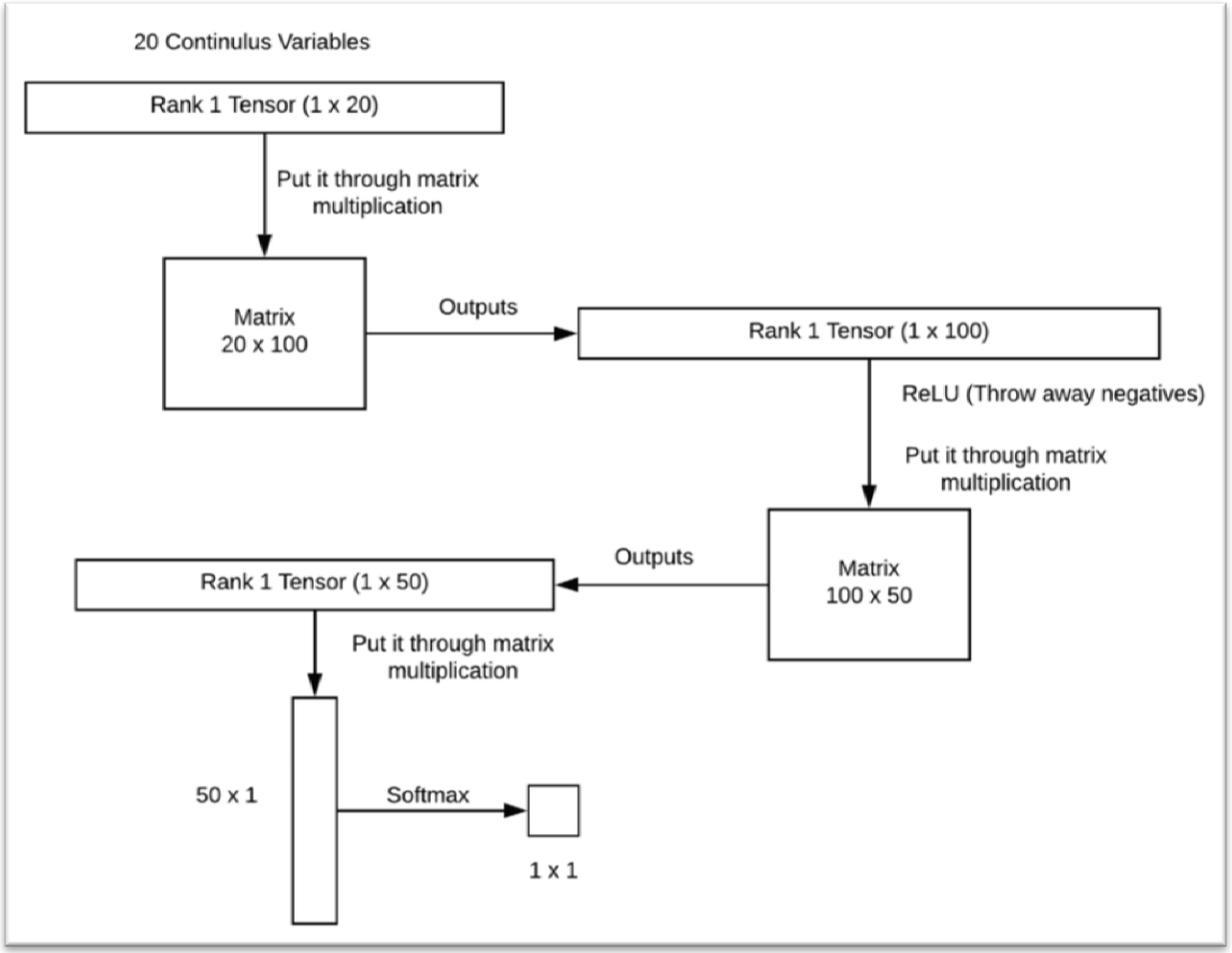
- 目的：销量预测，使用均方根百分比误差评估结果
- 首先创建模型数据对象，该对象包含一个验证集、训练集和可选测试集，之后调用lr_find, learn.fit等函数进行训练等。
- 这里的不同之处在于没有使用ImageClassifierData.from_csv或.from_paths，这里需要ColumnarModelData的模型数据，我们调用from_data_frame。

```
md = ColumnarModelData.from_data_frame(PATH, val_idx, df,
        yl.astype(np.float32), cat_flds=cat_vars, bs=128,
        test_df=df_test)
```

```
m = md.get_learner(emb_szs, len(df.columns)-len(cat_vars),
                   0.04, 1, [1000,500], [0.001,0.01],
                   y_range=y_range)
```

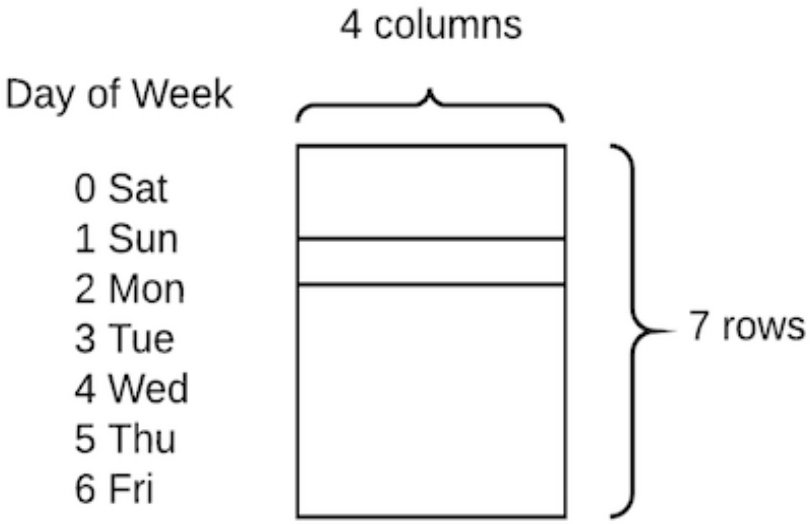
Continuous

- 对象：一个样本包含的所有连续变量（如温度、距离等）



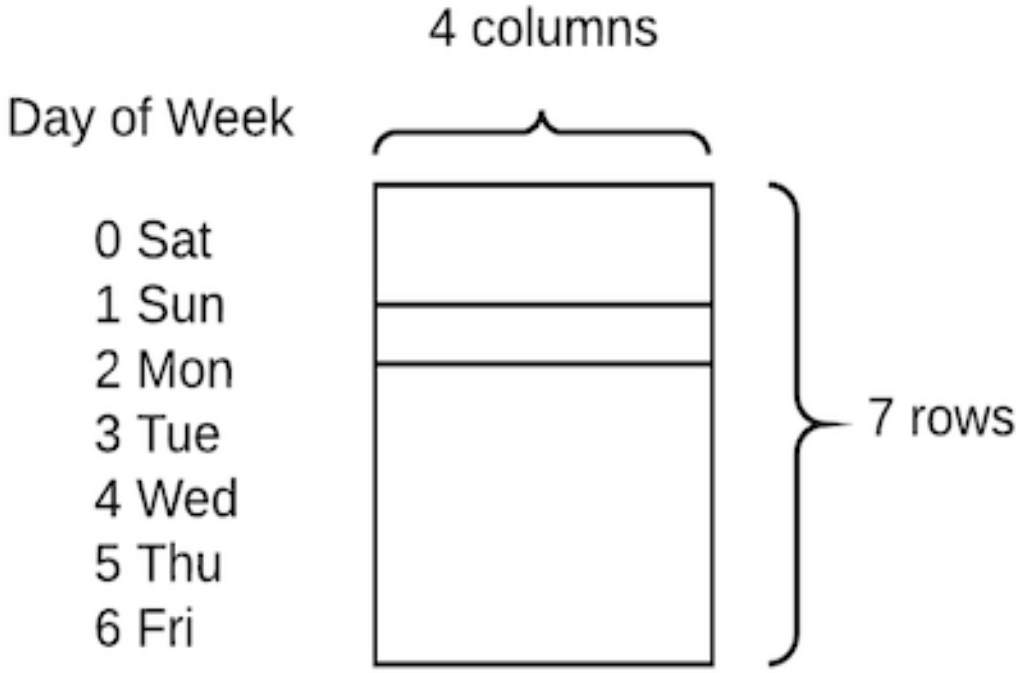
Categorical——嵌入矩阵

- 将非连续变量表示为一个连续变量的特征矩阵输入到神经网络中，如下图中将一周的每天分别表示为不同的4维向量
- 优点：相比(0,0,…1,0)形式的one-hot编码，这种embedding方法能够将单一的概念表现在多维空间，从而获得更丰富的语义概念，比如：在周末的行为与平日有所不同，则周六日的某些数字可能更高



Categorical——嵌入矩阵

- Tips：嵌入矩阵维数的经验值为基数大小除以2，但不大于50。（如：一个变量8个分类（8为基数），则取8*4的矩阵）



learner


```
m = md.get_learner(emb_szs, len(df.columns)-len(cat_vars), 0.04, 1,
                   [1000,500], [0.001,0.01], y_range=y_range)
lr = 1e-3
```

- Emb_szs：嵌入阵大小
- len(df.columns)-len(cat_vars)：数据框中连续变量的数量
- 0.04：嵌入阵自己的dropout值
- 1：最后一层输出数目
- [1000,500][0.001,0.01]：新添的第一二层输出数与dropout值

步骤总结

- 1.列出非连续变量和连续变量，并将它们放入pandas数据帧中
- 2.创建一个在验证集中需要的的行索引的列表
- 3.调用md = ColumnarModelData.from_data_frame(PATH, val_idx, df, yl.astype(np.float32), cat_flds=cat_vars, bs=128, test_df=df_test)
- 4.创建一个列表，列出每个嵌入矩阵的大小
- 5.调用get_learner
- 6.调用m.fit

lesson3-rossmann.ipynb报错

- 本例采用了数据分析的pandas库，0.22之后的版本可能会对summary报错，可以考虑将pandas版本降低至0.22后再运行
- 或者可以安装补丁包来解决此错误

  pip install -e git+https://github.com/mouradmourafiq/pandas-summary#egg=pandas-summary

目录 Contents

1 博文推荐

2 Overfitting与Dropout

3 结构化时间序列模型介绍

4 自然语言处理介绍

Nature Language Processing

- 例子：arXiv paper
- 通过学习十八个月内的arXiv中的论文，建立了一个语言模型，该模型可以根据给出的学科范畴（如计算机视觉），预测出这个学科的人会如何书写论文。
- 详细的示例可以参照代码***lang_model-arxiv.ipynb***

```
In [219]: sample_model(m,"<CAT> cscv <SUMM> algorithms that")

...use the same data to perform image classification are
 increasingly being used to improve the performance of image classification
 algorithms . in this paper , we propose a novel method for image classification
 using a deep convolutional neural network ( cnn ) . the proposed method is ...
```

IMDB影评情感判断——任务简介

- 目的：训练一个模型使之能判别IMDB上的影评是正面还是负面的
- 方法：先训练一个语言模型使之掌握影评书写的方式与规律，再以这个模型为基础去实施正面/负面的判别
- 数据集：来自IMDB的的50000条影评，为了方便训练，只使用了分化比较明显的数据集，即评分小于4和大于7的影评

IMDB影评情感判断—— 数据处理

- 数据预处理：

```
TEXT = data.Field(lower=True, tokenize=spacy_tok)
```

小写文本　　用spacy_tok标记

- 创建fast.ai数据模型对象：

```
bs=64; bptt=70

FILES = dict(train=TRN_PATH, validation=VAL_PATH, test=VAL_PATH)
md = LanguageModelData.from_text_files(PATH, TEXT, **FILES, bs=bs,
                                       bptt=bptt, min_freq=10)
```

bs：batch size

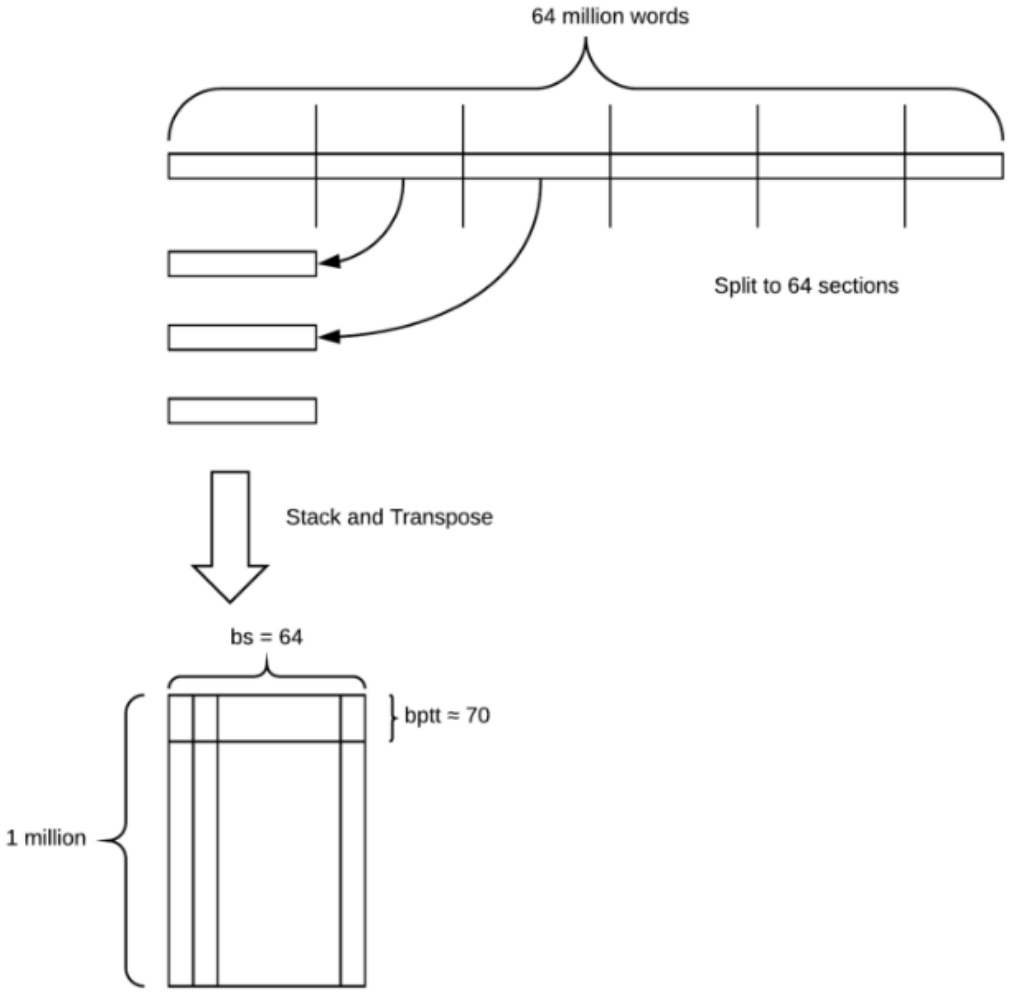
Text：torchtext的field定义

min_freq=10：将出现频率小于该值的词设为未知（unknown）

bptt：每一个batch的每一行中会处理多少词，增大该值会提高对于时间和存储量的要求，但相应的也会提高学习长句的能力。

理解batch和bptt

- 将所有词语连续排列后分为64段，再以左下角方式排列，batch里的每个数据取70个词
- batch里每个数据包含词的数量约等于bptt的值
- Tips：每轮都随机改变batch里每个数据包含词的数量，这样每一轮就会得到稍微不同的文本——类似于在计算机视觉任务中进行的shuffle。不能随机地打乱单词顺序，所以采用随机地移动文本断点的方法。



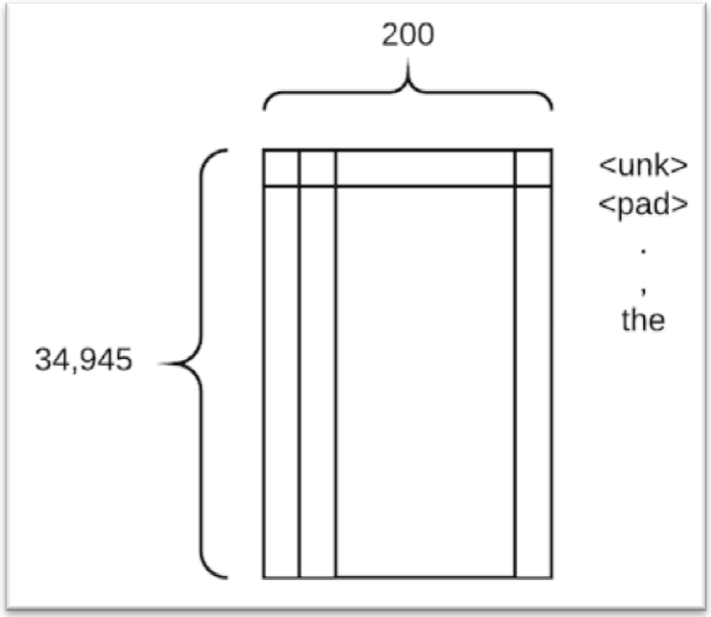
IMDB影评情感判断—— 数据处理

- 处理之后产生text.vocab,存储在文本中看到的唯一单词(或标记)，以及每个单词如何映射到唯一的整型id。

```
md.trn_ds[0].text[:12]

['i',
 'have',
 'always',
 'loved',
 'this',
 'story',
 '-',
 'the',
 'hopeful',
 'theme',
 ',',
 'the']

TEXT.numericalize([md.trn_ds[0].text[:12]])

Variable containing:
   12
   35
  227
  480
   13
   76
   17
    2
 7319
  769
    3
    2
```

IMDB影评情感判断——模型建立

- 本例中的text.vocab中共有34945个单词，这里选用的嵌入阵大小为200；在自然语言处理中，嵌入阵的大小一般为50~600之间
- 嵌入阵中的参数可以从头训练，也可以选取一些预先训练好的嵌入阵模型，如Word2vec 和 GloVe。



IMDB影评情感判断——语言模型训练

- 使用下面的语句建立模型并进行训练，其中动量、adam优化器以及几种dropout等概念会在后续课程讲到。

```
opt_fn = partial(optim.Adam, betas=(0.7, 0.99))
```

```
learner = md.get_model(opt_fn, em_sz, nh, nl, dropouti=0.05,
                       dropout=0.05, wdrop=0.1, dropoute=0.02,
                       dropouth=0.05)
learner.reg_fn = partial(seq2seq_reg, alpha=2, beta=1)
learner.clip=0.3
```

- 测试结果表明本课程所训练的模型可以有效学习imdb影评的书写规律并做出预测
- 最后采用我们训练好的模型为基础来构建情感倾向的预测

IMDB影评情感判断——构建情感判断网络

- 要使用预先训练过的模型，需要从语言模型中保存vocab，因为我们需要确保相同的单词映射到相同的id，并不再将所有影评前后相连

```
TEXT = pickle.load(open(f'{PATH}models/TEXT.pkl','rb'))
```

```
IMDB_LABEL = data.Field(sequential=False)
```

- 接着使用split定义torchtext数据集并创建一个modeldata对象：

```
splits = torchtext.datasets.IMDB.splits(TEXT, IMDB_LABEL, 'data/')

t = splits[0].examples[0]

t.label, ' '.join(t.text[:16])

('pos', 'ashanti is a very 70s sort of film ( 1979 , to be precise )
.')
```

```
md2 = TextData.from_splits(PATH, splits, bs)
```

IMDB影评情感判断——训练情感判断网络

- 调用get_model获取learner并在其中加载预先训练好的语言模型，同时使用分层学习率，并增加clipping的大小，以使SGDR更好地工作

```
m3 = md2.get_model(opt_fn, 1500, bptt, emb_sz=em_sz, n_hid=nh,
                   n_layers=nl, dropout=0.1, dropouti=0.4,
                   wdrop=0.5, dropoute=0.05, dropouth=0.3)

m3.reg_fn = partial(seq2seq_reg, alpha=2, beta=1)

m3.load_encoder(f'adam3_20_enc')
```

```
m3.clip=25.
lrs=np.array([1e-4,1e-3,1e-2])

m3.freeze_to(-1)
m3.fit(lrs/2, 1, metrics=[accuracy])
m3.unfreeze()
m3.fit(lrs, 1, metrics=[accuracy], cycle_len=1)

[ 0.       0.45074  0.28424  0.88458]

[ 0.       0.29202  0.19023  0.92768]
```

经过训练，该模型对于IMDB影评情感的判断可以达到94.5%的准确率。

lesson4-imdb.ipynb报错

- 需要调用spacy包，请在终端下载python -m spacy download en，否则会报错
- 出现如下错误，需在指定路径下新建models文件夹

```
pickle.dump(TEXT, open(f'{PATH}models/TEXT.pkl','wb'))
```

```
---------------------------------------------------------------------------
FileNotFoundError                         Traceback (most recent call last)
<ipython-input-25-1589b40f7022> in <module>()
----> 1 pickle.dump(TEXT, open(f'{PATH}models/TEXT.pkl','wb'))

FileNotFoundError: [Errno 2] No such file or directory: 'data/aclImdb/models/TEXT.pkl'
```

- 需要合理设置batch size和bptt值，否则可能会出现内存溢出的情况

谢谢！