实　　验　　报　　告

学　　号	20020006107	姓　　名	王义钧	专业班级	通信工程
课程名称	自然语言处理			学期	2022 年 春 季学期
任课教师	仲国强	完成日期	2022/5/27	上机课时间	2022/5/26
实验名称	自然语言处理的应用——基于 mindspore 的情感分类实验				

一、 实验目的

按照处理文本的粒度不同，情感分析可分为词语级、短语级、句子级、段落级以及篇章级等几个研究层次。这里以“段落级”为例，输入为一个段落，输出为影评是正面还是负面的信息。

本次实验，以 IMDB 影评情感分类体验 MindSpore 在自然语言处理上的应用。

二、 实验内容：

1. 准备环节。
2. 加载数据集，进行数据处理。
3. 定义网络。
4. 定义优化器和损失函数。
5. 使用网络训练数据，生成模型。
6. 得到模型之后，使用验证数据集，查看模型精度情况。

三、 实验过程：

1、预安装 Annaconda 和 mindspore

安装完成 Annaconda 并配置好环境变量。

进入 cmd 验证 conda 环境是否配置好。

验证过程遇到输入 conda activate 命令后无响应的错误：

```
CommandNotFoundError: Your shell has not been properly configured to use 'conda activate'.
To initialize your shell, run
$ conda init <SHELL_NAME>

Currently supported shells are:
  - bash
  - fish
  - tcsh
  - xonsh
  - zsh
  - powershell
See 'conda init --help' for more information and options.

IMPORTANT: You may need to close and restart your shell after running 'conda init'.
```

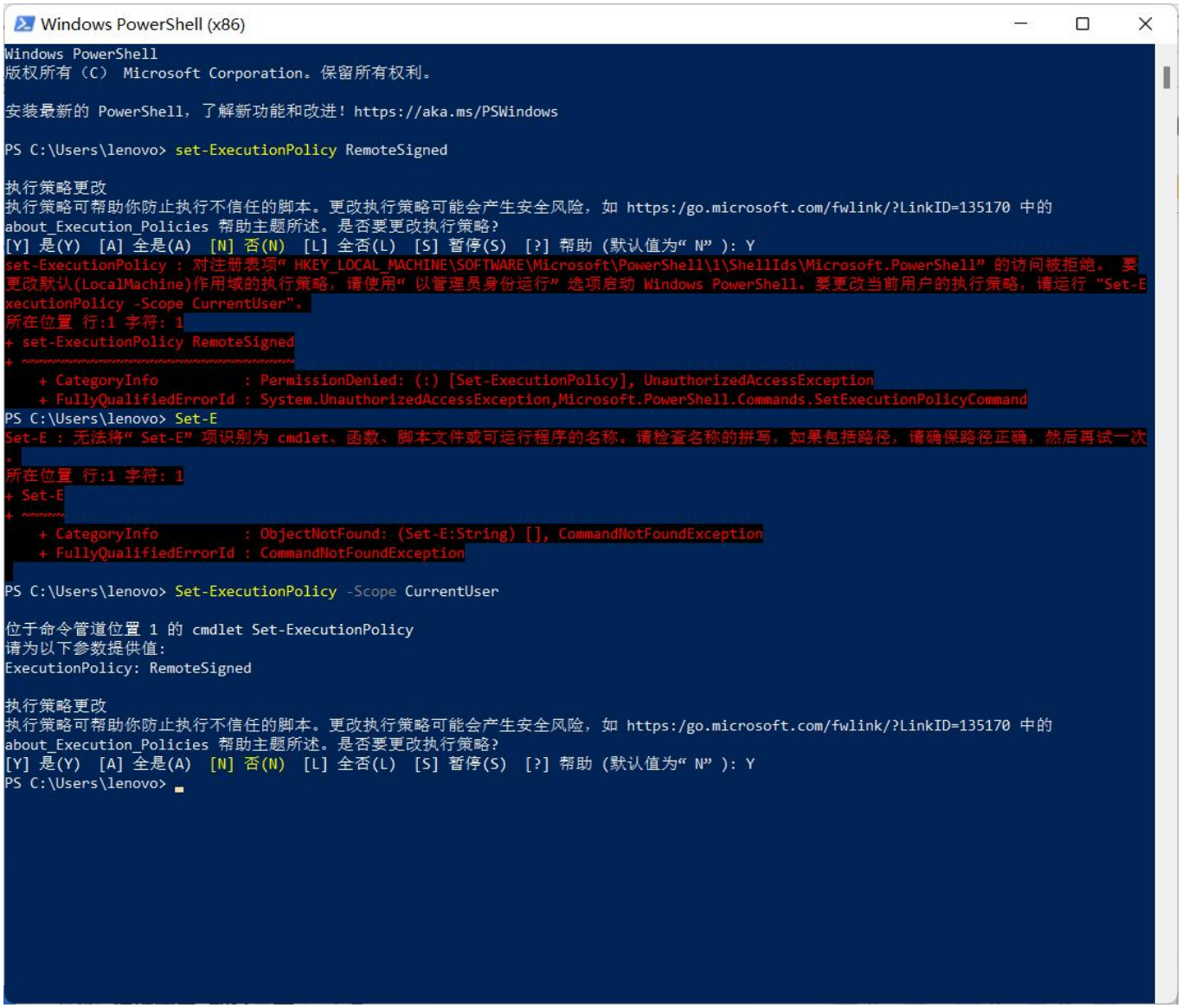
解决方法：

（1）通过 Win10 开始菜单进入 Windows PowerShell（管理员）

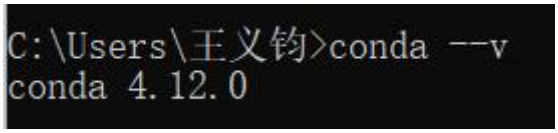
（2）输入命令 set-ExecutionPolicy RemoteSigned，如果无其他提示的红色错误后，输入 Y 直接解决。

（3）因为我的电脑受到管理员账户的控制，权限有限，我根据输入步骤 2 中的红色提示

信息后，输入以下命令行 Set-ExecutionPolicy -Scope CurrentUser ，然后在 ExecutionPolicy: 参数值这个位置输入 RemoteSigned 并最终也解决了问题。



结果：



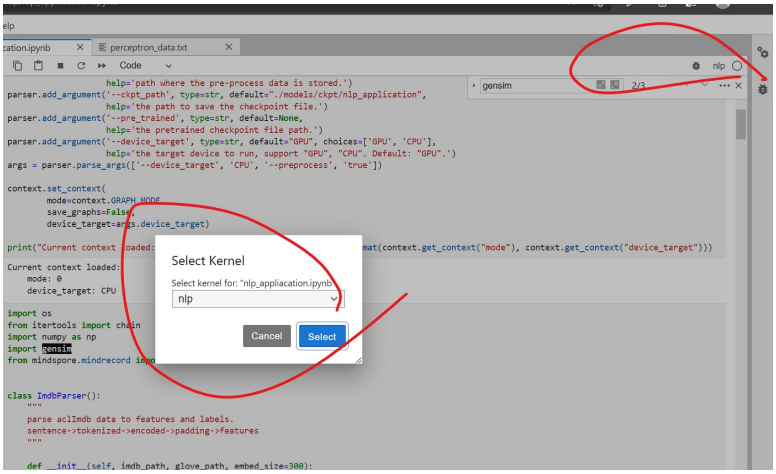
2、配置好 kernels 并启动 jupyter lab

在下面运行数据集训练代码时遇到错误，所以这里需要重新配置环境，并更换 jupyter 的核。

Pip env list 查看可用的核及其地址 并将不需要的核删去。

```
# conda environments:
#
base                  *  D:\Anaconda3
nlp                      D:\Anaconda3\envs\nlp
```

更改 jupyter lab 当前环境下的核。



3、测试运行训练集代码

①配置训练相关参数：

经过测试，一般没有问题：

[2]:
```
import argparse
from mindspore import context
from easydict import EasyDict as edict

# LSTM CONFIG
lstm_cfg = edict({
    'num_classes': 2,
    'learning_rate': 0.1,
    'momentum': 0.9,
    'num_epochs': 10,
    'batch_size': 64,
    'embed_size': 300,
    'num_hiddens': 100,
    'num_layers': 2,
    'bidirectional': True,
    'save_checkpoint_steps': 390,
    'keep_checkpoint_max': 10
})

cfg = lstm_cfg

parser = argparse.ArgumentParser(description='MindSpore LSTM Example')
parser.add_argument('--preprocess', type=str, default='false', choices=['true', 'false'], help='whether to preprocess da
parser.add_argument('--aclimdb_path', type=str, default="./datasets/aclImdb", help='path where the dataset is stored.')
parser.add_argument('--glove_path', type=str, default="./datasets/glove", help='path where the GloVe is stored.')
parser.add_argument('--preprocess_path', type=str, default="./preprocess", help='path where the pre-process data is stor
parser.add_argument('--ckpt_path', type=str, default="./models/ckpt/nlp_application", help='the path to save the checkpo
parser.add_argument('--pre_trained', type=str, default=None, help='the pretrained checkpoint file path.')
parser.add_argument('--device_target', type=str, default="GPU", choices=['GPU', 'CPU'], help='the target device to run,
args = parser.parse_args(['--device_target', 'CPU', '--preprocess', 'true'])

context.set_context(
    mode=context.GRAPH_MODE,
    save_graphs=False,
    device_target=args.device_target)

print("Current context loaded:\n mode: {}\n device_target: {}".format(context.get_context("mode"), context.get_context("
```

```
Current context loaded:
 mode: 0
 device_target: CPU
```

[3]:
```
import os
from itertools import chain
import numpy as np
import gensim
from mindspore.mindrecord import FileWriter
```

②预处理数据集：

调用 convert_to_mindrecord 函数执行数据集预处理。在该函数中，将数据集格式转换为 MindRecord 格式，便于 MindSpore 读取。

开始出现各种问题：

问题 1：未定义 args。

解决方法：该问题出现是因为我直接在 conda 终端上，运行 python 代码，将代码分割运行。args 继承的是第一段代码中的定义，而我单独运行第二段，因此查找不到。将其放到 jupyter lab 中，依次运行即可，或者在第二段代码中加入 import *.py。

问题 2：找不到对象 parse

```
def convert_to_mindrecord(embed_size, aclimdb_path, preprocess_path, glove_path):
    """ convert imdb dataset to mindrecoed dataset """
    parser = ImdbParser(aclimdb_path, glove_path, embed_size)
    parser.parse()

    if not os.path.exists(preprocess_path):
        print(f"preprocess path {preprocess_path} is not exist")
        os.makedirs(preprocess_path)

    train_features, train_labels, train_weight_np = parser.get_datas('train')
    _convert_to_mindrecord(preprocess_path, train_features, train_labels, train_weight_np)

    test_features, test_labels, _ = parser.get_datas('test')
    _convert_to_mindrecord(preprocess_path, test_features, test_labels, training=False)

if args.preprocess == "true":
    os.system("rm -f ./preprocess/aclImdb* weight*")
    print("============== Starting Data Pre-processing ==============")
    convert_to_mindrecord(cfg.embed_size, args.aclimdb_path, args.preprocess_path, args.glove_path)
    print("========================= Successful =========================")
```

```
============== Starting Data Pre-processing ==============
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
Input In [3], in <cell line: 175>()
    176 os.system("rm -f ./preprocess/aclImdb* weight*")
    177 print("============== Starting Data Pre-processing ==============")
--> 178 convert_to_mindrecord(cfg.embed_size, args.aclimdb_path, args.preprocess_path, args.glove_path)
    179 print("========================= Successful =========================")

Input In [3], in convert_to_mindrecord(embed_size, aclimdb_path, preprocess_path, glove_path)
    161 """ convert imdb dataset to mindrecoed dataset """
    162 parser = ImdbParser(aclimdb_path, glove_path, embed_size)
--> 163 parser.parse()
    165 if not os.path.exists(preprocess_path):
    166     print(f"preprocess path {preprocess_path} is not exist")

AttributeError: 'ImdbParser' object has no attribute 'parse'
```

解决方法：该问题的出现是因为 parse 函数在前面已经定义过了，但是我错误的将其缩进到第一个类中，导致后面无法调用。

问题 3：无效的 utf8 字符

```
============== Starting Data Pre-processing ==============
---------------------------------------------------------------------------
RuntimeError                              Traceback (most recent call last)
D:\Temp\ipykernel_21940\2911205663.py in <module>
    176     os.system("rm -f ./preprocess/aclImdb* weight*")
    177     print("============== Starting Data Pre-processing ==============")
--> 178     convert_to_mindrecord(cfg.embed_size, args.aclimdb_path, args.preprocess_path, args.glove_path)
    179     print("======================== Successful ========================")

D:\Temp\ipykernel_21940\2911205663.py in convert_to_mindrecord(embed_size, aclimdb_path, preprocess_path, glove_path)
    168
    169     train_features, train_labels, train_weight_np = parser.get_datas('train')
--> 170     _convert_to_mindrecord(preprocess_path, train_features, train_labels, train_weight_np)
    171
    172     test_features, test_labels, _ = parser.get_datas('test')

D:\Temp\ipykernel_21940\2911205663.py in _convert_to_mindrecord(data_home, features, labels, weight_np, training)
    156     writer.add_index(["id", "label"])
    157     writer.write_raw_data(data)
--> 158     writer.commit()
    159
    160 def convert_to_mindrecord(embed_size, aclimdb_path, preprocess_path, glove_path):

D:\Anaconda3\envs\wyj\lib\site-packages\mindspore\mindrecord\filewriter.py in commit(self)
    300         if not self._writer.get_shard_header():
    301             self._writer.set_shard_header(self._header)
--> 302         ret = self._writer.commit()
    303         if self._index_generator is True:
    304             if self._append:

D:\Anaconda3\envs\wyj\lib\site-packages\mindspore\mindrecord\shardwriter.py in commit(self)
    218             MRMCommitError: If failed to flush data to disk.
    219         """
--> 220         ret = self._writer.commit()
    221         if ret != ms.MSRStatus.SUCCESS:
    222             logger.error("Failed to commit.")

RuntimeError: [json.exception.type_error.316] invalid UTF-8 byte at index 10: 0xE9
```

解决方法：我一开始认为这是默认编码环境导致的问题，后面经查询验证发现是文件地址中含有中文。经更改后得以解决。

结果：

```
============== Starting Data Pre-processing ==============
======================== Successful ========================
```

此时文件夹下生成预处理文件 preprocess：

名称	修改日期	类型
acllmdb_test.mindrecord0	2022/5/27 16:39	MINDRECORD0
acllmdb_test.mindrecord0.db	2022/5/27 16:39	Data Base File
acllmdb_test.mindrecord1	2022/5/27 16:39	MINDRECORD1
acllmdb_test.mindrecord1.db	2022/5/27 16:39	Data Base File
acllmdb_test.mindrecord2	2022/5/27 16:39	MINDRECORD2
acllmdb_test.mindrecord2.db	2022/5/27 16:39	Data Base File
acllmdb_test.mindrecord3	2022/5/27 16:39	MINDRECORD3
acllmdb_test.mindrecord3.db	2022/5/27 16:39	Data Base File
acllmdb_train.mindrecord0	2022/5/27 16:39	MINDRECORD0
acllmdb_train.mindrecord0.db	2022/5/27 16:39	Data Base File
acllmdb_train.mindrecord1	2022/5/27 16:39	MINDRECORD1
acllmdb_train.mindrecord1.db	2022/5/27 16:39	Data Base File
acllmdb_train.mindrecord2	2022/5/27 16:39	MINDRECORD2
acllmdb_train.mindrecord2.db	2022/5/27 16:39	Data Base File
acllmdb_train.mindrecord3	2022/5/27 16:39	MINDRECORD3
acllmdb_train.mindrecord3.db	2022/5/27 16:39	Data Base File
weight.txt	2022/5/27 16:39	文本文档

③创建数据集并读取第一个 label 的数据列表：

创建数据集并读取第 1 个 batch 中的 label 数据列表，和第 1 个 batch 中第 1 个元素的 feature 数据。

经过测试，该步骤一般不会出现问题：

结果如下图所示：

```
The first batch contains label below:
[0 0 1 1 1 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 1 1 1 0 1 1 1 0 0 1 0 0 1 0 1
 0 0 0 0 1 0 0 1 1 1 0 0 0 1 1 1 1 0 1 0 0 1 1 0 1 1 0]
```

④定义网络：

导入初始化网络所需模块，定义模块来实现我们需要的网络参数及网络状态。然后针对 CPU 场景，自定义单层 LSTM 小算子堆叠，来实现多层 LSTM 大算子功能。实例化 SentimentNet ，创建网络，最后输出网络中加载的参数。

出现问题：nn.LSTMCell 已检测

```
PS D:\NLP\实验三\CNN> python DefineNet.py
Current context loaded:
  File "DefineNet.py", line 161, in <module>
    network = SentimentNet(vocab_size=embedding_table.shape[0],
  File "DefineNet.py", line 127, in __init__
    self.encoder = StackLSTM(input_size=embed_size,
  File "DefineNet.py", line 63, in __init__
    layers.append(nn.LSTMCell(input_size=input_size_list[i],
  File "D:\python\下载\lib\site-packages\mindspore\nn\layer\rnn_cells.py", line 73, in wrapper
    raise ValueError(f"The arguments of `nn.LSTMCell` from old MindSpore version(<1.6) are detected, "
ValueError: The arguments of `nn.LSTMCell` from old MindSpore version(<1.6) are detected, if you still need use single LSTM layer, please use `nn.LSTM` instead.
Exception ignored in: <function Cell.__del__ at 0x0000023CCDBEE700>
Traceback (most recent call last):
  File "D:\python\下载\lib\site-packages\mindspore\nn\cell.py", line 346, in __del__
    if self.compile_cache:
  File "D:\python\下载\lib\site-packages\mindspore\nn\cell.py", line 337, in __getattr__
    raise AttributeError("The '{}' object has no attribute '{}'.".format(type(self).__name__, name))
AttributeError: The 'LSTMCell' object has no attribute 'compile_cache'.
```

解决方法：按照报错提示，将 nn.LSTMCell 更改为 nn.LSTM 即可。

网络定义结果：

```
OrderedDict([('embedding.embedding_table', Parameter (name=embedding.embedding_table, shape=(252193, 300), dtype=Float32, requires_grad=False)),
('encoder.weight0', Parameter (name=encoder.weight0, shape=(320800, 1, 1), dtype=Float32, requires_grad=True)), ('encoder.weight1', Parameter (name
=encoder.weight1, shape=(240800, 1, 1), dtype=Float32, requires_grad=True)), ('decoder.weight', Parameter (name=decoder.weight, shape=(2, 400), dty
pe=Float32, requires_grad=True)), ('decoder.bias', Parameter (name=decoder.bias, shape=(2,), dtype=Float32, requires_grad=True))])
```

⑤加载训练集并保存模型：

加载训练数据集并配置好 checkpoint 信息。然后使用 model.train 接口，进行模型训练。这个阶段耗时较长，我是利用 CPU 跑的，速度大概在 4.1GHz 左右，每一步的时长却高达 2500ms。总运行时长 4 个小时。

遇到问题：weight_ih_l0 已存在。

```
============== Starting Training ==============
WARNING: Logging before InitGoogleLogging() is written to STDERR
[CRITICAL] PARSER(16980,1,?):2022-5-27 10:2:32 [mindspore\ccsrc\pipeline\jit\parse\resolve.cc:158] ResolveParameterObj] The parameter construct_wrapper.1:weight_ih_l0 , its name 'weight_ih_l0' already exists. Please set a unique name for the parameter.
Traceback (most recent call last):
    self.compile(*inputs)
  File "D:\python\下载\lib\site-packages\mindspore\nn\cell.py", line 937, in compile
    _cell_graph_executor.compile(self, *inputs, phase=self.phase, auto_parallel_mode=self._auto_parallel_mode)
  File "D:\python\下载\lib\site-packages\mindspore\common\api.py", line 1006, in compile
    result = self._graph_executor.compile(obj, args_list, phase, self._use_vm_mode())
RuntimeError: mindspore\ccsrc\pipeline\jit\parse\resolve.cc:158 ResolveParameterObj] The parameter construct_wrapper.1:weight_ih_l0 , its name 'weight_ih_l0' already exists. Please set a unique name for the parameter.
```

解决方法：该文件我无法定位在哪里，可能是某个代码中定义的变量。因为是 mindspore 中的问题，我又重新安装了一遍 mindspore，重开了一个新的文件夹继续跑，发现可以解决这个问题。

训练集训练结果：

```
============== Starting Training ==============
epoch: 1 step: 78, loss is 0.6957096
epoch: 1 step: 156, loss is 0.6743079
epoch: 1 step: 234, loss is 0.6446719
epoch: 1 step: 312, loss is 0.65861005
epoch: 1 step: 390, loss is 0.69689345
epoch time: 1215301.586 ms, per step time: 3116.158 ms
epoch: 2 step: 78, loss is 0.70962024
epoch: 2 step: 156, loss is 0.6102886
epoch: 2 step: 234, loss is 0.6964881
epoch: 2 step: 312, loss is 0.65087044
epoch: 2 step: 390, loss is 0.66379094
epoch time: 1041371.743 ms, per step time: 2670.184 ms
epoch: 3 step: 78, loss is 0.62043875
```

```
epoch: 6 step: 390, loss is 0.2953758
epoch time: 983130.995 ms, per step time: 2520.849 ms
epoch: 7 step: 78, loss is 0.4075886
epoch: 7 step: 156, loss is 0.2930706
epoch: 7 step: 234, loss is 0.3777066
epoch: 7 step: 312, loss is 0.3998994
epoch: 7 step: 390, loss is 0.345225
epoch time: 8633244.550 ms, per step time: 22136.524 ms
epoch: 8 step: 78, loss is 0.38182682
epoch: 8 step: 156, loss is 0.4456376
epoch: 8 step: 234, loss is 0.291192
epoch: 8 step: 312, loss is 0.30604628
epoch: 8 step: 390, loss is 0.27665028
epoch time: 971882.619 ms, per step time: 2492.007 ms
epoch: 9 step: 78, loss is 0.33856133
epoch: 9 step: 156, loss is 0.27251458
epoch: 9 step: 234, loss is 0.3823075
epoch: 9 step: 312, loss is 0.265261
epoch: 9 step: 390, loss is 0.29062575
epoch time: 968556.788 ms, per step time: 2483.479 ms
epoch: 10 step: 78, loss is 0.28647646
epoch: 10 step: 156, loss is 0.25654155
epoch: 10 step: 234, loss is 0.31627426
epoch: 10 step: 312, loss is 0.28853276
epoch: 10 step: 390, loss is 0.27155113
epoch time: 952513.996 ms, per step time: 2442.344 ms
============== Training Success ==============
```

⑥模型验证：

创建并加载验证数据集（ ds_eval ），加载由训练保存的 CheckPoint 文件，进行验证，查看模型质量。

最终验证得，对文本的情感分析正确率大概在 85%左右，基本达到理想的成果。

```
============== Starting Testing ==============
============== {'acc': 0.8562501282051282} ==============
```

四、 结果展示：

```
    return self._executor(args_list, phase)
KeyboardInterrupt
PS D:\NLP\lab3\RNN> python .\lab03.py
Current context loaded:
    mode: 0
    device_target: CPU
The first batch contains label below:
[0 0 1 1 1 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0 0 0 1 1 1 0 1 1 1 0 0 1 0 0 1 0 1
 0 0 0 0 1 0 0 1 1 1 0 0 0 1 1 1 1 0 1 0 0 1 1 0 1 1 0]

The feature of the first item in the first batch is below
OrderedDict([('embedding.embedding_table', Parameter (name=embedding.embedding_table, shape=(252193, 300), dtype=Float32
, requires_grad=False)), ('encoder.weight0', Parameter (name=encoder.weight0, shape=(320800, 1, 1), dtype=Float32, requi
res_grad=True)), ('encoder.weight1', Parameter (name=encoder.weight1, shape=(240800, 1, 1), dtype=Float32, requires_grad
=True)), ('decoder.weight', Parameter (name=decoder.weight, shape=(2, 400), dtype=Float32, requires_grad=True)), ('decod
er.bias', Parameter (name=decoder.bias, shape=(2,), dtype=Float32, requires_grad=True))])
============== Starting Training ==============
epoch: 1 step: 78, loss is 0.6957096
epoch: 1 step: 156, loss is 0.6743079
epoch: 1 step: 234, loss is 0.6446719
epoch: 1 step: 312, loss is 0.65861005
epoch: 1 step: 390, loss is 0.69689345
epoch time: 1215301.586 ms, per step time: 3116.158 ms
epoch: 2 step: 78, loss is 0.70962024
epoch: 2 step: 156, loss is 0.6102886
epoch: 2 step: 234, loss is 0.6964881
epoch: 2 step: 312, loss is 0.65087044
epoch: 2 step: 390, loss is 0.66379094
epoch time: 1041371.743 ms, per step time: 2670.184 ms
epoch: 3 step: 78, loss is 0.62043875
```

```
epoch: 6 step: 390, loss is 0.2953758
epoch time: 983130.995 ms, per step time: 2520.849 ms
epoch: 7 step: 78, loss is 0.4075886
epoch: 7 step: 156, loss is 0.2930706
epoch: 7 step: 234, loss is 0.3777066
epoch: 7 step: 312, loss is 0.3998994
epoch: 7 step: 390, loss is 0.345225
epoch time: 8633244.550 ms, per step time: 22136.524 ms
epoch: 8 step: 78, loss is 0.38182682
epoch: 8 step: 156, loss is 0.4456376
epoch: 8 step: 234, loss is 0.291192
epoch: 8 step: 312, loss is 0.30604628
epoch: 8 step: 390, loss is 0.27665028
epoch time: 971882.619 ms, per step time: 2492.007 ms
epoch: 9 step: 78, loss is 0.33856133
epoch: 9 step: 156, loss is 0.27251458
epoch: 9 step: 234, loss is 0.3823075
epoch: 9 step: 312, loss is 0.265261
epoch: 9 step: 390, loss is 0.29062575
epoch time: 968556.788 ms, per step time: 2483.479 ms
epoch: 10 step: 78, loss is 0.28647646
epoch: 10 step: 156, loss is 0.25654155
epoch: 10 step: 234, loss is 0.31627426
epoch: 10 step: 312, loss is 0.28853276
epoch: 10 step: 390, loss is 0.27155113
epoch time: 952513.996 ms, per step time: 2442.344 ms
============== Training Success ==============
============== Starting Testing ==============
============== {'acc': 0.8562501282051282} ==============
```

五、 心得体会：

该实验过程中遇到了很多问题，耗时也比较长，但通过尝试独立解决这些问题，我进一步理解了利用 mindspore 进行自然语言处理应用上的体验，并且理解了如何通过定义和初始化基于 LSTM 的 SentimentNet 网络进行模型训练及验证正确率。自己动手解决问题的能力得到了进一步提升，对 jupyter lab 的编写与实现过程也有了更加深入的理解。