**文件编号：3107-SWC2018-20180045**

**受控状态：■受控 □非受控**

**保密级别：□公司级 □部门级 ■项目级 □普通级**

**采纳标准：CMMI DEV V1.2**

****

**图美集**

**Temage**

**技术研究报告**

**Version 1.1.1**

**2019.3.1**

**Written by 3107**

****

**All Rights Reserved**

目录

[1 引言 1](#_Toc2343412)

[1.1 编写目的 1](#_Toc2343413)

[1.2 项目概述 1](#_Toc2343414)

[1.3 项目背景 1](#_Toc2343415)

[1.4 术语和缩略语 1](#_Toc2343416)

[1.5 参考资料 2](#_Toc2343417)

[2 问题聚焦 2](#_Toc2343418)

[2.1 问题描述 2](#_Toc2343419)

[2.2 问题抽象 3](#_Toc2343420)

[2.3 问题定位 3](#_Toc2343421)

[2.4 问题评估 3](#_Toc2343422)

[2.5 问题分解 3](#_Toc2343423)

[3 相关工作 4](#_Toc2343424)

[4 技术方案 4](#_Toc2343425)

[4.1 技术方向 4](#_Toc2343426)

[4.2 模型选择 4](#_Toc2343427)

[4.2.1 模型设计 4](#_Toc2343428)

[4.2.2 模型结构 6](#_Toc2343429)

[4.2.3 数据集 6](#_Toc2343430)

[4.3 结果期望 7](#_Toc2343431)

[5 技术实践 7](#_Toc2343432)

[5.1 使用的深度学习框架及依赖的Library 7](#_Toc2343433)

[5.2 模型训练过程 7](#_Toc2343434)

[5.2.1 英特尔AI DevCloud环境配置 7](#_Toc2343435)

[5.2.2 模型训练过程及结果 8](#_Toc2343436)

[5.3 模型验证过程 9](#_Toc2343437)

[6 结果验证 9](#_Toc2343438)

记录更改历史

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **序号** | **更改原因** | **版本** | **作者** | **更改日期** | **备 注** |
| 1 | 创建 | 1.0.0 | 队员B | 2018/11/02 | 初步阅览文档并进行一些部分初始化 |
| 2 | 创建 | 1.0.1 | 队员B | 2018/11/05 | 撰写相关工作和技术方案 |
| 3 | 更新 | 1.0.2 | 队员B | 2018/11/10 | 撰写浏览推荐部分文档 |
| 4 | 更新 | 1.0.3 | 队员B | 2018/11/13 | 强调突出前后端工作 |
| 5 | 更新 | 1.1.0 | 队员B | 2019/02/13 | 修改技术报告初赛部分 |
| 6 | 更新 | 1.1.1 | 队员B | 2019/02/15 | 添加复赛部分内容 |
| 7 | 更新 | 1.2.0 | 队员B | 2019/03/20 | 丰富相关描述 |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

# 引言

## 编写目的

该项目技术研究报告的编写目的是为了全面深入分析和介绍本次项目的技术细节从项目的背景，到项目的整体框架设计，以及最终的实现细节，Temage不断深入，层次分明的展现项目技术全貌。

该技术开发文档重点介绍了项目的技术架构和技术细节，对本项目使用的模型进行详细的阐述，对用于训练的数据集进行说明。

## 项目概述

Temage使用tensorflow、keras等深度学习框架在后端进行推断和tesorflow.js和keras.js在前端进行推断，合理安排模型的分布，在本地浏览器中放置模型以承担一定量计算任务，减少服务端的运转负载及降低网络延迟，对于需要大量知识库和语料库且模型较大的功能，Temage使用服务端进行推断。

## 项目背景

随着互联网时代的到来，互联网媒体逐渐抢占传统媒体市场，尤其是近几年的自媒体的崛起，使得传播主体多样化，平民化，普泛化。现在的网络用户只需要实名认证就可以在微博，微信公众号等自媒体平台上展现自我。因此，图文结合的使用领域越来越多，自媒体运营者为了吸引更多的用户，在文章内容和形式上绞尽脑汁。本项目希望设计一款使用深度学习技术的web应用，为用户提供个性化的图文结合和文本编辑服务，并以长图或其他格式发布到各大平台。

图片识别在近些年有巨大的发展，在ILSVRC 2012中，Alex Krizhevsky基于GPU实现了有60million参数的模型——AlexNet，赢得了比赛的第一名。这个工作是开创性的，它引领了接下来ILSVRC的风潮。随后几年中，Google，Baidu等大公司也加入到其中，得到了错误率更低的模型。同时，深度学习在自然语言处理中也大展身手。2013年，Google开源了一款用于词向量计算的工具——word2vec，引起了工业界和学术界的关注。随后提出的RNN，LSTM更是大展身手，TextCNN在情感分析等方面更是有着令人惊叹的效果。

在本项目中，Temage将结合深度学习中图片识别和自然语言处理这两个部分，为用户提供具有优良性能的图文结合、智能排版编辑功能。

## 术语和缩略语

[1] Tensorflow： Tensorflow是一个采用数据流图(data flow graphs),用于数值计算的开源软件库。

[2] Keras： Keras是一个高层神经网络API，Keras由纯Python编写而成并基Tensorflow、Theano以及CNTK后端。

[3] Tornado: Tornado 是一种 Web 服务器软件的开源版本。Tornado 和现在的主流 Web 服务器框架（包括大多数 Python 的框架）有着明显的区别：它是非阻塞式服务器，而且速度相当快。

[4] Django: Django 是 Python 编程语言驱动的一个开源模型-视图-控制器（MVC）风格的 Web 应用程序框架。

## 参考资料

[1] Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition[J]. arXiv preprint arXiv:1409.1556, 2014.

[2] Peters M E, Neumann M, Iyyer M, et al. Deep contextualized word representations[J]. arXiv preprint arXiv:1802.05365, 2018.

[3] Lai S, Xu L, Liu K, et al. Recurrent Convolutional Neural Networks for Text Classification[C]//AAAI. 2015, 333: 2267-2273.

[4] Zeman D, Hajič J, Popel M, et al. CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies[J]. Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies, 2018: 1-21.

[5] Fares M, Kutuzov A, Oepen S, et al. Word vectors, reuse, and replicability: Towards a community repository of large-text resources[C]//Proceedings of the 21st Nordic Conference on Computational Linguistics, NoDaLiDa, 22-24 May 2017, Gothenburg, Sweden. Linköping University Electronic Press, 2017 (131): 271-276.

[6] Wang L, Li Y, Huang J, et al. Learning two-branch neural networks for image-text matching tasks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, 41(2): 394-407.

# 问题聚焦

## 问题描述

基于设想，Temage有三个主要问题需要解决：

1. 如何根据用户输入给予用户风格推荐
2. 如何将用户的图片和文字进行匹配并合理排版
3. 如何推荐相关内容使得用户可以从中得到借鉴

## 问题抽象

根据问题描述提出的问题，Temage进一步抽象问题

针对问题一，Temage既需要从用户的文本中提取特征，又需要根据用户使用历史，向用户提供风格进行选择。

针对问题二，Temage需要从图像和文本中分别提取特征，并将两者进行比较，从而进行匹配，通过匹配进行文本和图片的融合。可以抽象为三个部分，第一部分是将图片映射到向量空间，第二部分是将文字映射到向量空间，第三部分将图片和文本两个维度的向量进行匹配。

针对问题三，Temage需要对用户作品按关键词进行分类，在用户进行搜索时给用户提供相关作品的展示。

## 问题定位

本项目中的技术问题主要为自然语言处理和图像识别方面的问题。

## 问题评估

对于问题一，文本的特征提取在自然语言处理领域中，是较为成熟的一个部分，可供选择的模型较多，普适性高；另一方面，基于时序性操作的推断，可以使用LSTM神经网络模型。

对于问题二，图像识别在计算机视觉领域也是较为成熟的一个部分，众多的团队提供了非常多深度学习的模型可供挑选和fine-tuning，文本嵌入也是近几年提出的风靡全球的模型，从word2vec到doc2vec以及最新的ELMo，结合RNN、LSTM，Temage有成熟的方案能够解决这个问题，有关于两方面向量匹配的问题，《Learning Two-Branch Neural Networks for

Image-Text Matching Tasks》对网络结构有详尽的描述，可供Temage参考。

对于问题三，Temage对比相似性的依据应当是文本中的关键词和问题二中推断出的风格主题。最终结合问题一中的解决方案，关键词中添加分类名称，Temage使用关键词匹配来进行推荐。

## 问题分解

问题一可分解为文本分类和用户习惯追踪两个子问题。文本分类问题，可以使用较为成熟的模型进行训练，推断。用户习惯追踪Temage可以使用LSTM神经网络模型。结合两个部分，给予用户最终的风格推荐。

问题二较为复杂，可分解为图像识别，文本嵌入和图文匹配三个问题。图像识别问题可以使用是使深度学习在众多机器学习算法中脱颖而出的 CNN 模型，Temage借鉴VGG16的网络结构并适当缩小。文本嵌入问题可以使用RNN-LSTM对文中单词或句子进行encode，得到表示单词或句子的向量。对于图文匹配问题，Temage可以基于余弦计算等方法找到找到最为匹配的图片与文字，对文章进行排版。

问题三有成熟的解决方案，Temage对作品文本提取关键词，进行比对，为提高搜索效率，Temage可以使用ElasticSearch开源搜索引擎框架对数据库建立索引。

# 相关工作

1. VGG卷积神经网络是牛津大学在2014年提出来的模型。当这个模型被提出时，由于它的简洁性和实用性，马上成为了当时最流行的卷积神经网络模型。它在图像分类和目标检测任务中都表现出非常好的结果。在2014年的ILSVRC比赛中，VGG 在Top-5中取得了92.3%的正确率。
2. ELMo于2018年2月由AllenNLP提出，与word2vec或GloVe不同的是其动态词向量的思想，其本质即通过训练language model，对于一句话进入到language model获得不同的词向量。根据实验可得，使用了ELMo词向量之后，许多NLP任务都有了大幅的提高。
3. 2015年，中科院在AAAI上发表了一篇名为《Recurrent Convolutional Neural Networks for Text Classification》的论文，并提出了Recurrent Convolutional Neural Network(RCNN)的模型用于在时间内进行文本分类。在论文提及的数据集中效果均优于原有的分类模型。
4. 2019年，Svetlana Lazebnik在IEEE Transactions on Pattern Analysis and Machine Intelligence，发表《Learning Two-Branch Neural Networks for Image-Text Matching Tasks》一文，就关于解决图像和文字匹配问题的双分支网络作出了详细的阐述，并给出了相关数据集的测试结果。

# 技术方案

## 技术方向

本项目中将对文本进行处理，所使用的模型中将会用到RNN(LSTM)、CNN（TextCNN）。对图像的处理，Temage将会使用CNN。

## 模型选择

### 模型设计

对于文本分类问题，Temage决定使用ELMo对单词进行embedding，之后使用RNN(LSTM)对文章进行embedding，之后使用TextCNN进行文本分类。

对于用户习惯追踪，Temage使用成熟的LSTM神经网络建立模型进行训练。

对于图文匹配问题，Temage使用双分支网络——使用CNN对图像进行卷积，输出向量，将向量和ELMo+LSTM模型产出的单句embedding进行比对，得到最佳匹配。

为分担服务器工作，提高用户体验感，Temage将文本分类所使用的TextCNN模型放置在用户前端，并使用tensorflow.js等前端深度学习框架让模型在前端进行推断。

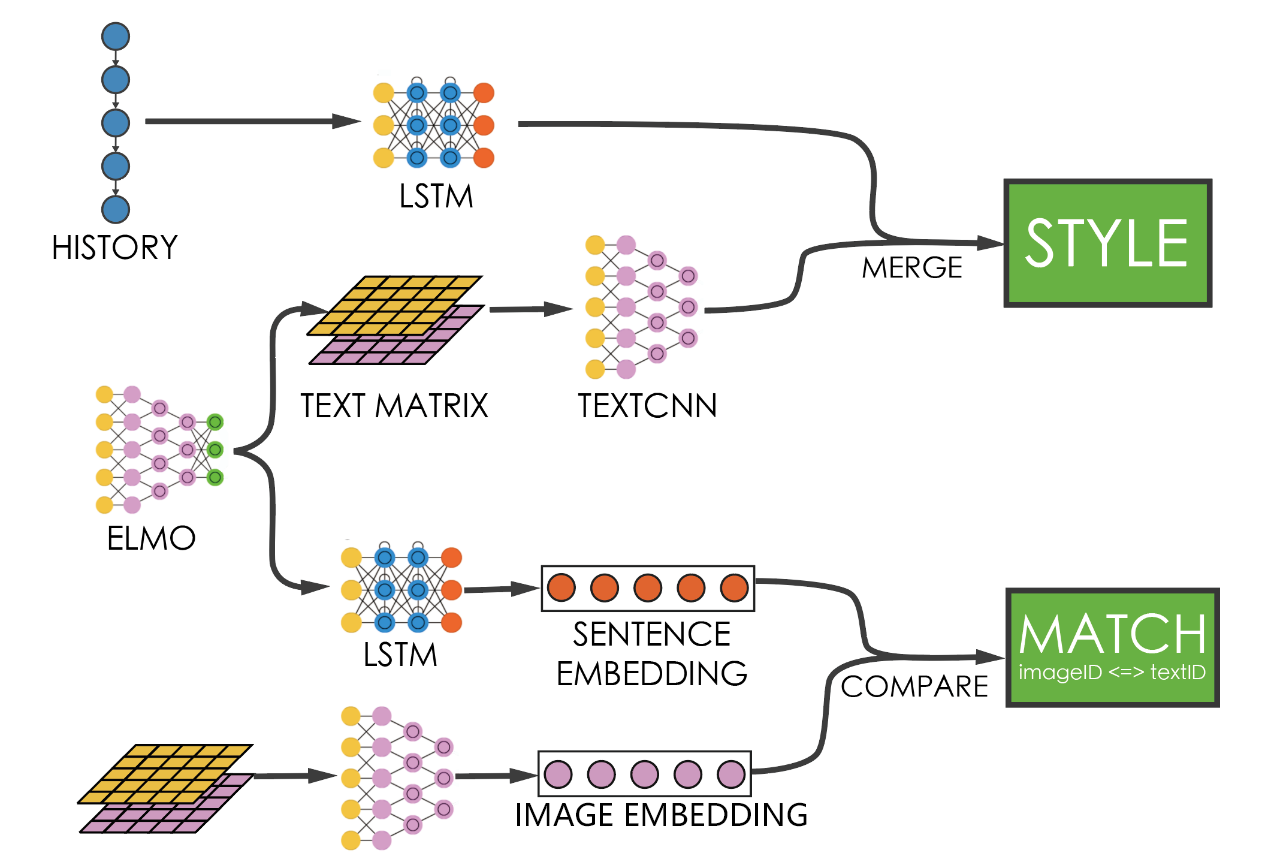


图 4.1 模型设计图

Temage使用ELMO的原因主要在于ELMO考虑了一词多义的问题（Deep contextualized word representations），并且解决了使用word2vec中的无法创建集外词（OOV）问题。

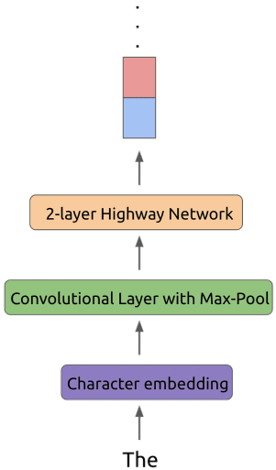
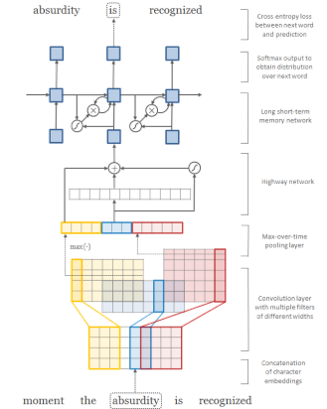


图4.2 ELMo利用sub-word + CNN，在character的层次上获取集外词的词义

### 模型结构

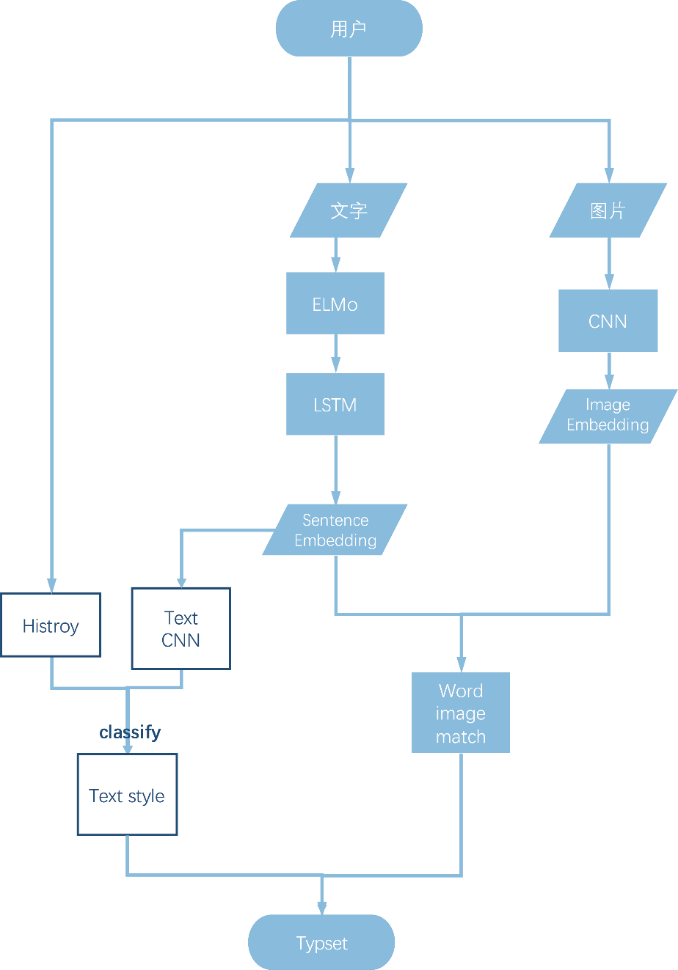
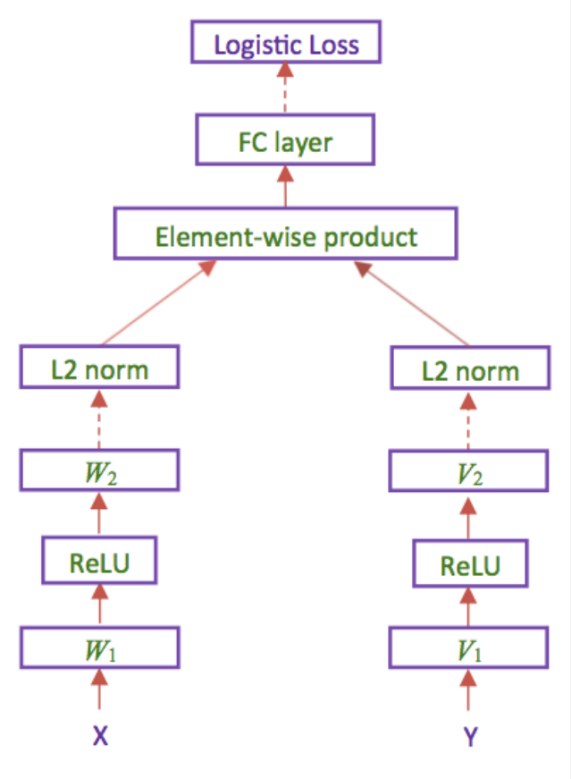


图4.3 模型流程图，其中白色代表用户浏览器中执行的流程，蓝色代表服务端执行的流程



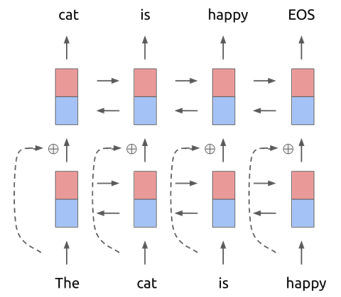


图4.4模型结构图，其中左图为双分支模型的整体结构，右图为ELMo模型结构图

### 数据集

1. 文本分类模型：THUCNews数据集。（http://thuctc.thunlp.org/）THUCNews是根据新浪新闻RSS订阅频道的历史数据筛选过滤生成，包含74万篇新闻文档（2.19 GB），均为UTF-8纯文本格式，分布于14个分类中，用于文本分类进行风格推荐的模型训练。训练时采取sklearn的sklearn.model\_selection.train\_test\_split来分割数据集，取20%作为测试集，80%作为训练集。凭借此Temage将得到新闻中的正文，上述数据可用于文本分类的训练。
2. 用户习惯追踪模型：哔哩哔哩数据集，共约140万篇文章以及120万个图文匹配样例（剔除无效文章之后）。使用爬虫在哔哩哔哩专栏（https://www.bilibili.com/read/home）上爬取用户的文章发布历史，使用训练好的文本分类模型对文章进行文本分类，将结果用于用户习惯追踪模型的训练。
3. 图文匹配模型：哔哩哔哩数据集，共约140万篇文章以及120万个图文匹配样例（剔除无效文章之后）。训练时采取sklearn的sklearn.model\_selection.train\_test\_split来分割数据集，取20%作为测试集，80%作为训练集。凭借此Temage将得到新闻中的图片和图片上下文，从图片上下文结合ELMo模型，得到与图片对应的向量嵌入。（Temage将爬虫得到的数据用作正例，分别使用同文乱序和非同文乱序两种方式产生反例，正例与反例的比例为6:4）
4. ELMo模型Temage准备fine-tuning HIT-SCIR/ELMoForManyLangs

## 结果期望

对于文本分类模型，基于TextCNN的网络较为成熟，预计的分类成功率在90%以上。

对于用户习惯追踪模型，基于LSTM的网络较为成熟，预计的预测成功率在90%以上。

对于图文匹配模型，基于双分支模型，根据Svetlana Lazebnik的相关工作报告，预计模型的匹配成功率在70%左右。

# 技术实践

## 使用的深度学习框架及依赖的Library

Temage的模型一共分为两个部分——Temage Trained Model和 Fine-tuning Model。

Temage Trained Model：

|  |  |
| --- | --- |
| 名称 | 用途 |
| keras | 构建、训练、保存文本分类模型 |
| pandas | 训练和测试数据的读取和基本预处理 |
| jieba | 对文本进行分词 |
| sklearn | 划分训练集和测试集，进行K-fold验证 |
| numpy | 数据预处理 |
| matplotlib | 绘制训练过程图像 |

Fine-tuning Model:

|  |  |
| --- | --- |
| pytorch | 构建和调用ELMo模型 |
| codecs | 编码转换 |

框架和library具体版本详见项目代码.

## 模型训练过程

### 英特尔AI DevCloud环境配置

Intel DevCloud上的具体环境配置为Keras 2.2.4, Tensorflow 1.10.0。

训练时需要在DevCloud服务器上的.bash\_profile中配置几个特定的环境变量（CC/ LD\_LIBRARY\_PATH）才能将工作提交到集群中正常运行，具体配置可见提交代码中的.bash\_profile文件。

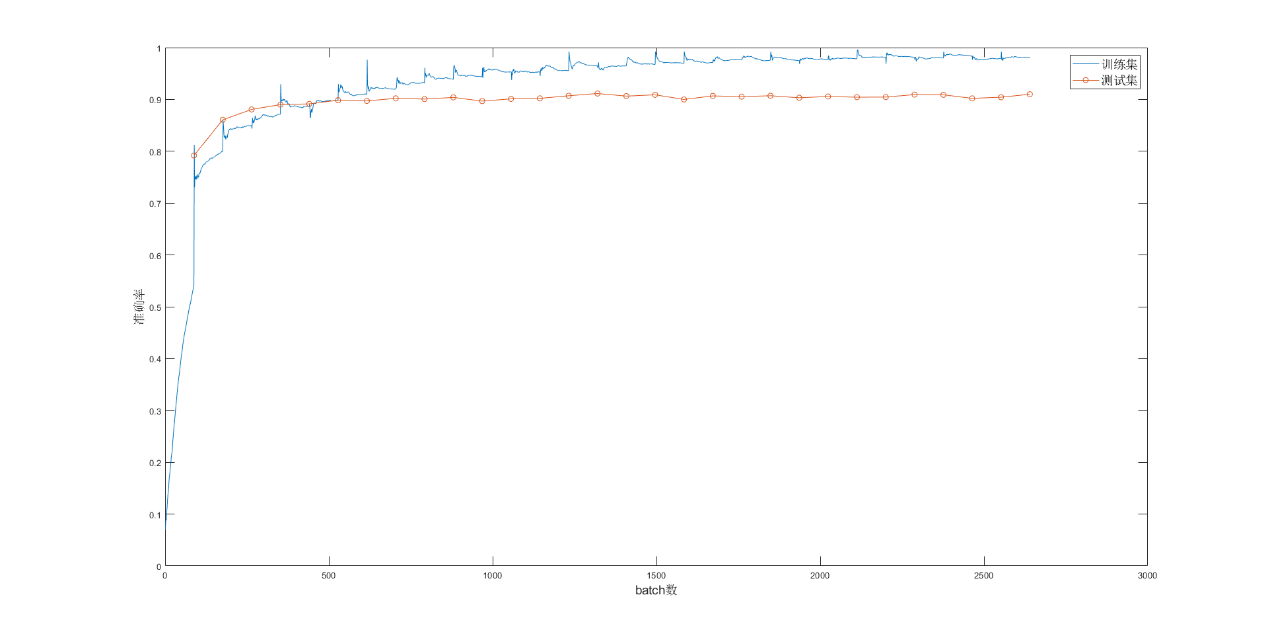
### 模型训练过程及结果

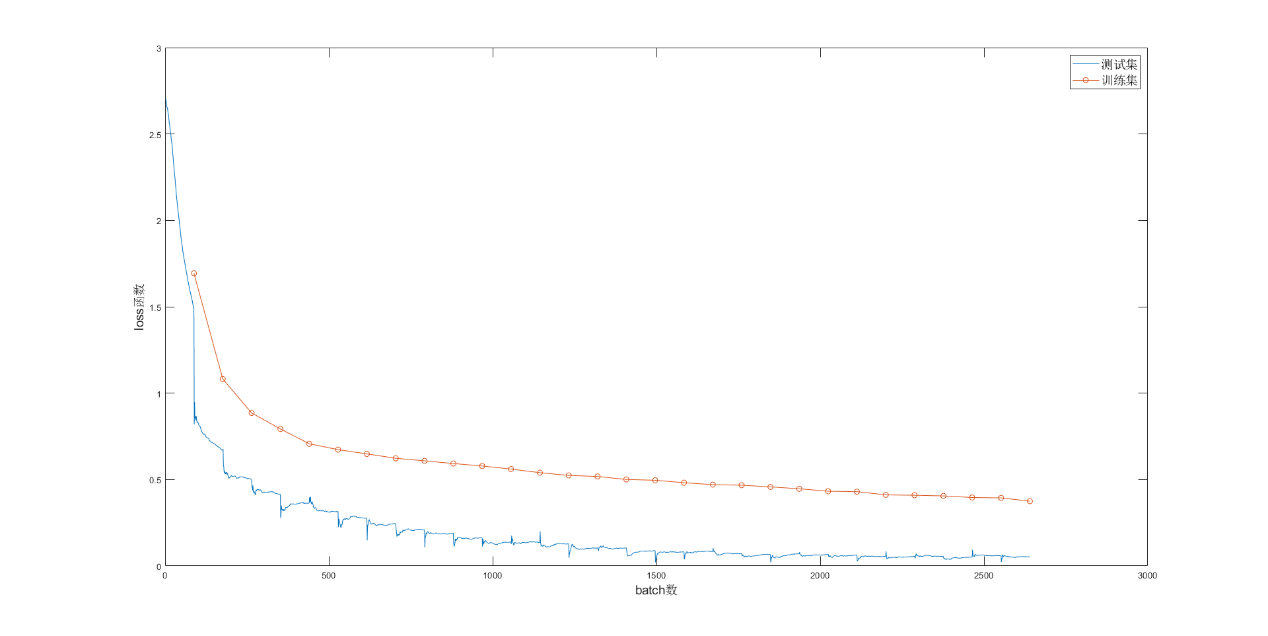
1. 文本分类模型在Intel DevCloud上进行训练，其余模型在校方提供的服务器上进行训练。

文本分类模型在14个类别，总共14000篇文章的数据集上进行训练。在batch\_size=128,

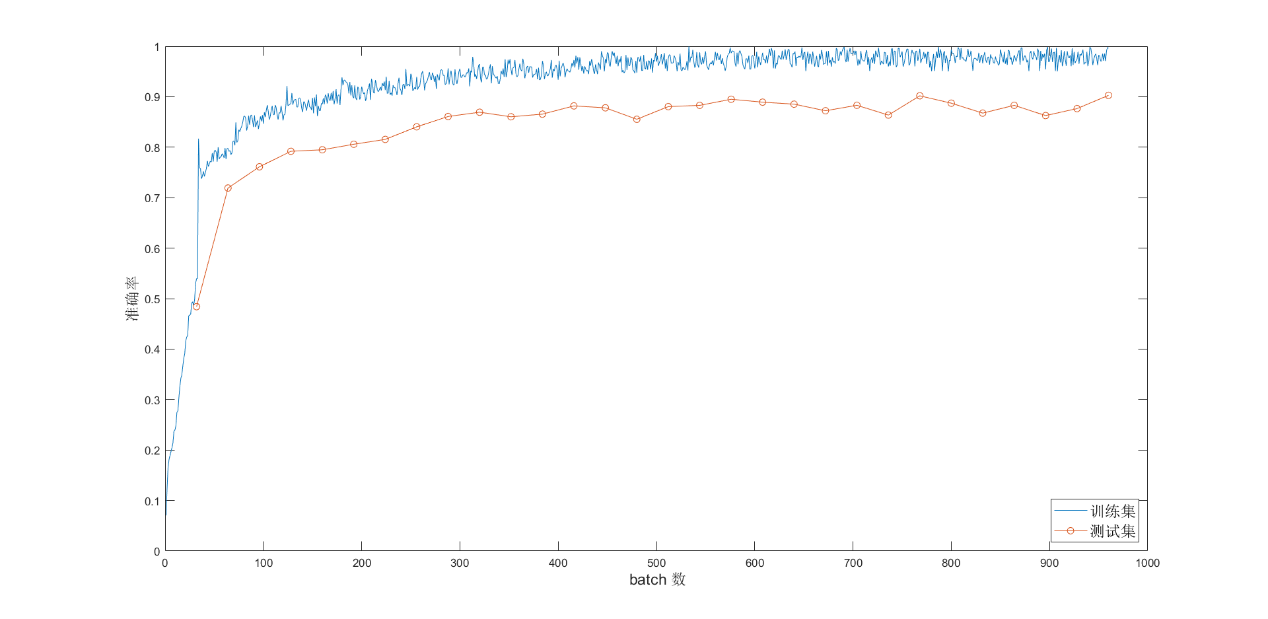
epoch=30的参数设置下，训练集准确率、测试集准确率、交叉熵（categorical crossentropy）loss函数如下图所示。

（其中测试集的准确率和交叉熵loss函数的值均为每个epoch取一次，训练集的准确率和交叉熵loss函数的值为每个batch取一次，下同）

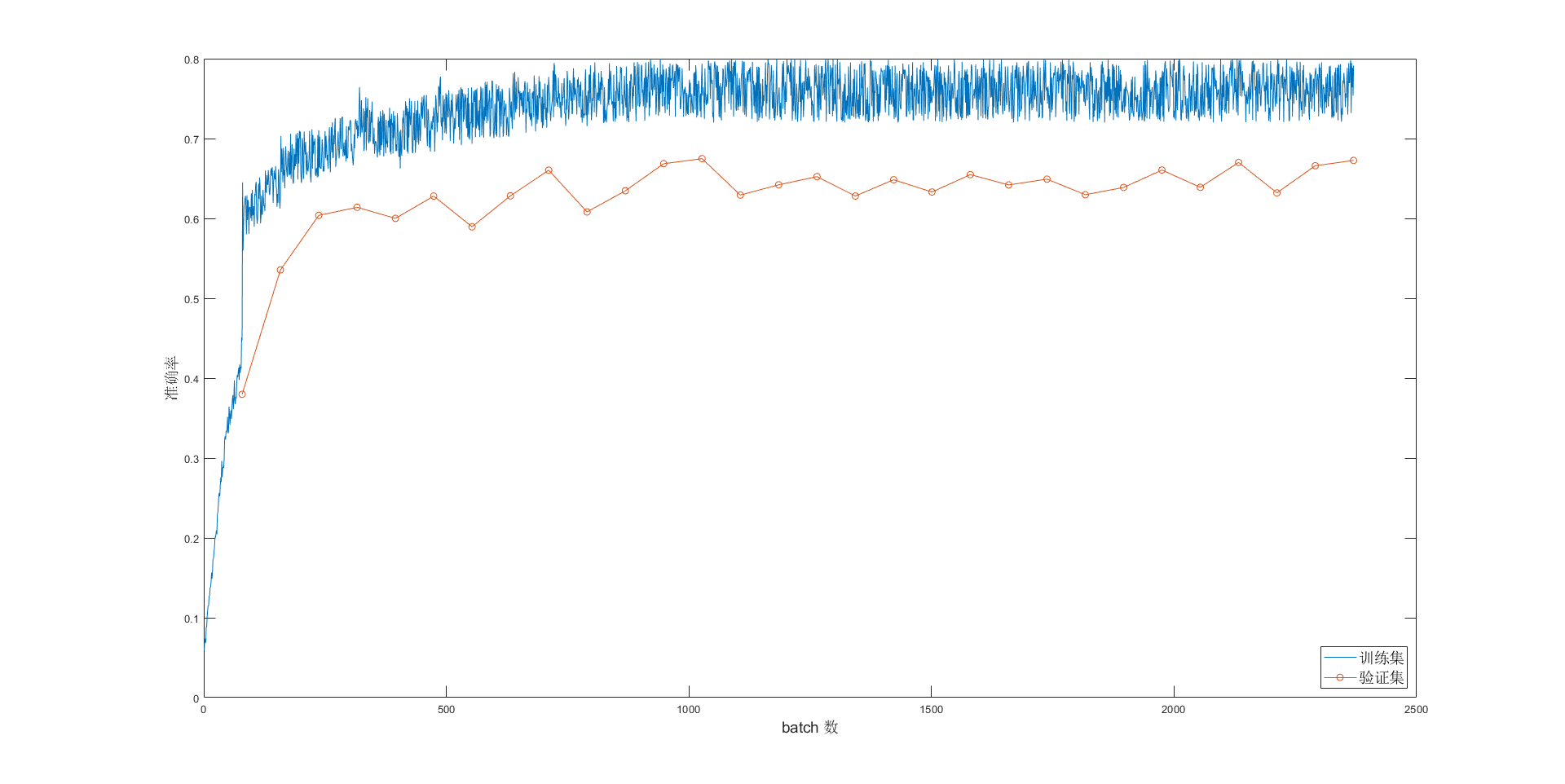




1. 用户习惯追踪模型在2000个用户，总共12000篇文章的数据集上进行训练。在batch\_size=64, epoch=30的参数设置下，训练集、测试集的准确率如下图所示。



1. 图文匹配模型在总共10000个样本的数据集（正确的图文匹配为6000个，生成的错误图文匹配为4000个）上进行训练。在batch\_size=128, epoch=30的参数设置下，训练集、测试集的准确率如下图所示。



## 模型验证过程

Temage的各模型均以K-Fold 交叉验证为基础进行验证，将模型平均分成K组，取其中一组作为测试集，其余作为训练集。交叉验证有效利用了有限的数据，并且评估结果能够尽可能接近模型在测试集上的表现，可以作为模型优化的指标使用。以下为Temage各模型在K-Fold交叉验证下的准确率表现。

# 结果验证

1. 文本分类模型

文本分类模型使用K-Fold的变种Stratified Fold，K = 4

|  |  |
| --- | --- |
| 文本分类模型 | |
| 序号（K=1, 2, … , 10） | 准确率 |
| 1 | 0.9064285714285715 |
| 2 | 0.9057142857142857 |
| 3 | 0.9092857139451164 |
| 4 | 0.8907142853736878 |
| 5 | 0.8792857142857143 |
| 6 | 0.8778571428571429 |
| 7 | 0.9007142857142857 |
| 8 | 0.9021428571428571 |
| 9 | 0.8985714282308306 |
| 10 | 0.9078571428571428 |

1. 用户习惯追踪模型

标准K Fold，K=5

|  |  |
| --- | --- |
| 用户习惯追踪模型 | |
| 序号（K=1, 2, … , 10） | 准确率 |
| 1 | 0.8740692013062071 |
| 2 | 0.8932131938918028 |
| 3 | 0.8939332892892509 |
| 4 | 0.9019431670347694 |
| 5 | 0.8846428074871655 |

1. 图文匹配模型

|  |  |
| --- | --- |
| 用户习惯追踪模型 | |
| 序号（K=1, 2, … , 10） | 准确率 |
| 1 | 0.6526076873985585 |
| 2 | 0.6715381492804736 |
| 3 | 0.6556305024311877 |
| 4 | 0.6415483940511476 |
| 5 | 0.6514943187078461 |

综上所述，Temage的各个模型都有着相对稳定的表现。