

分类号: TP391.1

单位代码: 10058



天津工业大学  
TIANGONG UNIVERSITY

# 工程硕士学位论文

论文题目: 基于生成式对抗网络的食谱生成技术研究

工程领域: 计算机技术

学习方式: ☒全日制攻读 ☐非全日制攻读

作者姓名: 白艳

学校导师: 荣垂田

企业导师: 张仁杰

完成日期: 2020-12-21

# 独 创 性 声 明

本人声明所呈交的学位论文是本人在导师指导下进行的研究工作和取得的研究成果，除了文中特别加以标注和致谢之处外，论文中不包含其他人已经发表或撰写过的研究成果，也不包含为获得天津工业大学或其他教育机构的学位或证书而使用过的材料。与我一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

学位论文作者签名：

签字日期： 年 月 日

## 摘要

随着时代的变化，人们的生活水平逐渐提高，饮食习惯发生了翻天覆地的变化。在过去，大部分人在饮食上只是追求吃饱，只有少数人关注饮食的搭配与口味等。而现在，越来越多的人在饮食上有了更高的要求。中国是一个拥有众多人口的国家，其地域广阔，不同地区的饮食习惯不尽相同。为了丰富我国食谱多样性，满足人们日益增长的饮食需求，我们需要在现有食谱数据的基础上进行研究，然后准确、科学地挖掘到食物搭配的内在规律，创造新的食谱。

现如今，生成式对抗网络（GAN）引起了广泛关注，并在许多应用场景中取得了很好的效果。GAN 模型在逐步改进中，很多科研人员在原始 GAN 模型的基础上，通过创新，开发出各种 GAN 模型的变体，例如 WGAN、LSGAN、BEGAN 和 StackGAN 等。这些变体适用于解决各种场景中遇到的实际问题，展示了 GAN 家族的强大能力，也为我们研究食谱的生成提供了新思路。

在本文中，我们尝试应用 SeqGAN 框架来实现食谱生成任务。我们首先通过爬虫技术在网站上获取中文食谱数据，然后训练 SeqGAN 模型，使其可以生成接近真实的食谱文本。为了扩充食谱数据的样本数量、提高模型的泛化能力，我们采用文本数据增强技术，希望获得更好的训练效果。为了验证 SeqGAN 模型的食谱生成效果，我们对比了 SeqGAN 模型和 RNN 模型生成的食谱。

此外，我们考虑为食谱增加标签以满足人们的个性化需求，标签表示食谱的口味特征，例如“酱香味”、“酸甜味”和“香辣味”等。此时需要训练条件-生成式对抗网络（CGAN）。在 CGAN 中，我们将食谱数据的口味标签作为指导食谱生成的条件信息。通过分析食谱口味与食物之间的搭配关系，我们可以发现口味与食物（主要是辅料）之间的内部联系。

实验结果表明，本文训练的两种生成模型都能够通过学习食物之间的搭配关系来实现食谱的生成，这为人们在日常生活中进行食物搭配提供了科学的手段。

**关键词：**食物搭配；食谱；生成式对抗网络；条件-生成式对抗网络；卷积神经网络；长短期记忆网络

## ABSTRACT

With the changes of the times, people's living standards have gradually improved, and their eating habits have changed drastically. In the past, most people just wanted to eat enough and only a few people paid attention to the food pairing and tastes. Now, more and more people have higher requirements on recipes. China is a country with a large population and vast area, and the eating habits of different regions are not the same. In order to enrich the diversity of our country's recipes and meet people's growing dietary needs, we need to conduct research on the basis of existing recipes, and then accurately and scientifically explore the internal laws of food pairing and create new recipes.

Nowadays, Generative Adversarial Networks (GAN) has attracted widespread attention and has achieved good results in many applications. The GAN model is still being gradually improved. On the basis of the original GAN model, many researchers have developed many different GAN variants, such as WGAN, LSGAN, BEGAN, StackGAN, etc. These variants are suitable for solving practical problems encountered in various scenarios, which demonstrate the powerful capabilities of the GAN's family and provide new ideas for us to study the generation of recipes.

In this article, we try to apply the SeqGAN framework to achieve the task of recipes generation. We first obtain Chinese recipes from the website through crawler technology, and then train the SeqGAN model so that it can generate realistic recipes. In order to expand the number of recipes data and improve the generalization ability of the model, we use text-based data enhancement technology and hope to obtain a better training effect. In order to verify the recipes generation effect of the SeqGAN model, we compared the recipes generated by the SeqGAN model and the RNN model.

In addition, we are considering adding labels to recipes to meet people's personal needs. The labels indicate the taste characteristics of the recipes, such as "sauce flavor", "sour and sweet and flavor " and "spicy flavor". At this time, a training Condition Generative Adversarial Network (CGAN) is required. In CGAN, we use the taste labels of the recipes data as the condition information to guide the recipes generation. By analyzing the relationship between recipe tastes and ingredients, we

can find the internal connection between tastes and ingredients (mainly minor ingredients).

The experimental results show that the two generative models trained in this article can realize recipes generation by learning the collocation relationship between foods, which provides a scientific means for people to choose ingredients in daily life.

**Keywords:** Food pairing; Recipes; GAN; CGAN; CNN; LSTM

# 目 录

第一章 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外的研究现状.....	3
1.2.1 食谱数据处理相关研究.....	3
1.2.2 生成式对抗网络相关研究.....	5
1.3 本文的主要研究内容.....	8
1.4 本文的组织结构.....	9
第二章 研究基础和技术.....	11
2.1 数据采集技术.....	11
2.1.1 网络爬虫技术.....	11
2.1.2 爬虫框架.....	12
2.2 文本处理技术.....	15
2.2.1 文本数据预处理.....	15
2.2.2 文本表示技术.....	16
2.2.3 数据增强技术.....	17
2.3 深度学习技术.....	17
2.3.1 深度学习概念.....	17
2.3.2 深度学习框架介绍.....	17
2.3.3 卷积神经网络.....	18
2.3.4 循环神经网络.....	19
2.3.5 生成式对抗网络.....	20
2.3.6 条件-生成式对抗网络 .....	21
2.4 本章小结.....	21
第三章 基于生成式对抗网络的食谱生成.....	23
3.1 基于生成式对抗网络的食谱生成模型框架.....	23
3.2 基于生成式对抗网络的食谱生成模型算法流程.....	24
3.2.1 模型预训练.....	24
3.2.2 生成器训练.....	24
3.2.3 判别器训练.....	25
3.3 实验前的准备工作.....	27

3.3.1 数据采集.....	27
3.3.2 数据清洗.....	28
3.3.3 食谱成分处理.....	29
3.3.4 模型评价指标.....	29
3.4 基于生成式对抗网络的食谱生成实验.....	31
3.4.1 实验过程说明.....	31
3.4.2 实验结果及分析.....	33
3.5 基于生成式对抗网络和循环神经网络的食谱生成算法对比实验.....	35
3.5.1 基于循环神经网络的食谱生成算法.....	35
3.5.2 实验过程说明.....	35
3.5.3 实验结果及分析.....	35
3.6 本章小结.....	37
第四章 基于条件-生成式对抗网络的食谱生成 .....	39
4.1 条件对食谱生成结果的影响.....	39
4.2 基于条件-生成式对抗网络的食谱生成模型框架 .....	40
4.3 基于条件-生成式对抗网络的食谱生成算法 .....	41
4.3.1 食谱成分及标签处理.....	41
4.3.2 基于 LSTM 的生成器.....	42
4.3.3 基于 CNN 的判别器 .....	42
4.4 基于条件-生成式对抗网络的食谱生成实验 .....	43
4.4.1 实验过程说明.....	43
4.4.2 实验结果及分析.....	44
4.5 本章小结.....	45
第五章 总结与展望.....	47
5.1 总结.....	47
5.2 展望.....	47
参考文献.....	49
发表论文和参加科研情况.....	53
致谢.....	55

## 第一章 绪论

### 1.1 研究背景及意义

在我国，食物一直都是一个最基本、最普遍的话题，中国上下五千年的历史，为我们孕育了种类丰富、独具特色的饮食文化。正所谓“民以食为天”，食物的获取、选择与烹饪等方式，一直深刻地影响着我们日常生活的方方面面。我国地域广阔、人口众多，这种特性使得不同的食物有着不同的象征意义以及地域特色。我国的八大菜系享誉中外，各大菜系在选材、配料和烹饪等技艺方面，经过长期的演变而蕴含了独特的地方色彩。众所周知，我国菜系大致可分为八类，分别是川菜（四川）、鲁菜（山东）、苏菜（江苏）、粤菜（广东）、浙菜（浙江）、闽菜（福建）、湘菜（湖南）、徽菜（安徽）。不同的菜系体现着我国不同地区的文化和习俗。

随着我国经济的快速发展和人们生活水平的大幅改善，人们对于饮食有了新的要求，而不再简单地满足于一直使用过去的饮食配方，食谱需要推陈出新的难题越来越需要解决，以满足人们不断增长的新的需求。

然而在过去，由于食谱数据众多且分散在各地，大到城市、小到农村，网络技术也不够发达，仅凭人工难以准确、完整地收集、统计和整理，而且相应的数据采集技术发展迟滞，所以我们难以获取到高质量的食谱数据集，无法很好地开展食谱数据的研究分析工作。

近年来，随着互联网的发展，各种社交网站逐渐在人们的生活中流行起来，人们越来越习惯于在这些社交网络上分享自己在日常生活中的食物，包括这些美食的菜名、烹饪所需的食物名字及数量、烹饪时间、烹饪步骤、口味、功能等信息。逐渐地，各种以食谱制作、分享与交流美食为目的的社区网站越来越多。这些美食网站的出现，也吸引着越来越多的人参与到食谱的分享与交流中来，经过日积月累，我国食谱网站的规模越来越大，食谱种类越来越丰富。例如一个名叫“美食杰”的在线食谱社区网站，该社区网站通过美食菜谱吸引用户，是全网美食达人的聚集地。从 2007 年成立到发展至今，该网站将食物、健康、科技这三者相结合，已经拥有百万篇视频、图文菜谱，而且每年有 1 亿多次的网站浏览量，有 5,000 多万的正在接受并享用他们提供的烹饪指导。该社区网站帮助几千万的用户在日常的餐饮环境中，快速而又准确地了解自身所需的饮食配方，逐渐完善我国饮食系统的升级。



这些社区网站作为一种新型的人与人之间的交流方式,网站上面包含了非常丰富的内容,解决了以往食谱数据资源严重匮乏的困难,为我们提供了丰富的食谱数据,为开展关于食物搭配、食谱生成的研究提供了真实可靠的数据来源,为我们的研究结果提供了准确的、科学的、可靠的数据依据。这些数据用于挖掘、分析和研究,使我们从中获益匪浅<sup>[1]</sup>。基于提取的食谱社区网站的数据,我们可以使用先进的自然语言处理技术以及深度学习模型,分析食物之间的内在联系,找到食物之间的搭配规律。通过这些研究,我们可以通过训练生成网络,生成大量的新食谱,丰富中国现有的食谱种类,改善人们的饮食水平,为人们的食物选择提供新的可能。

同时,标签在处理海量文本、组织和管理结构化文本信息资源等方面显示出了不可替代的优越性,因此,标签广泛运用在了很多社交 APP 上,如微博、抖音、绿洲等。在微博上,任何用户在编辑某条微博之时,都可以加入一个或多个相应的标签,这些标签与该条微博的主要内容密切相关,代表了这条微博的主要内容。在企业中,他们可以为自己的产品内容加上标签,根据其标签内容匹配有意向的用户,然后将自己的产品推送给有兴趣的用户。由此可见,标签功能逐渐被广泛应用到信息筛选任务中,具有很高的研究意义和价值<sup>[2]</sup>。

对于“美食杰”食谱社区网站而言,标签可以指示食谱烹饪方式的难易程度,例如“新手尝试”、“初级入门”、“中级掌勺”和“高级厨师”等。标签也可以指示食谱的烹饪工艺,例如“蒸”、“煎”、“炒”、“炸”和“烘焙”等。标签还可以指示食谱的口味,例如“香辣味”、“酸甜味”、“酱香味”、“咖喱味”和“姜汁味”等。此外,标签可以指示食谱的食疗功效,例如功能性调理的“健脾开胃”标签、疾病调理的“骨质疏松”标签、脏腑调理的“气血双补”标签等。这些标签的应用,不仅可以帮助网站的开发者分类整理食谱数据、满足不同用户的个性化需求,还可以帮助网站用户快速定位到自己所需的食谱。我们可以尝试在生成网络中加入食谱数据的某种标签,将这些标签作为条件数据,用于指导生成模型生成有方向的食谱数据。

综上所述,食谱社区网站的迅速兴起与流行,为本文研究中国食谱的生成提供了数量足够、质量可靠的数据来源,我们通过训练生成式对抗网络,使其从现有的食谱中学习到不同食物之间的内在联系,可以开发新食谱、丰富食谱多样性。此外,为生成式对抗网络添加标签,基于特定条件生成食谱,可以满足不同人群的个性化需求。本文的研究具有重要的实际意义。

## 1.2 国内外的研究现状

在过去,国内的食谱数据收集困难,国外的食谱数据集对于我国饮食习惯的研究作用甚微,因此国内对于食谱生成的研究受到了限制。近年来,随着以用户为中心、以数据为基础、以食谱分享为主题的用户饮食管理服务平台的兴起,食谱社区网站吸引了几千万人的关注,食谱的数据量呈指数式增加,这些变化支持着越来越多的人开始尝试对食谱进行不同方面的科学研究,例如对食谱营养成分的研究和基于食谱内容的个性化推荐等。

同时,标签的分类检索功能使其在互联网的应用越来越常态化。对于食谱数据而言,不同的标签分别代表着食谱的不同特性。这些特性都是爱好美食、热爱烹饪的用户在日常生活中重点关注的内容。基于不同的标签,人们可以关注到同一食谱数据的不同属性。

目前,国内外关于食谱生成和生成式对抗网络有很多的研究,接下来,我们对这两部分的国内外研究现状分别展开论述。

### 1.2.1 食谱数据处理相关研究

受限于食谱数据收集困难,国内对于食物搭配和食谱生成方面的研究一直是一个尚待开发的领域,截止到目前,能找到的有价值的相关研究为数不多。近些年,随着先进的研究技术的发展,人们不再局限于传统的研究方式,开始将统计分析、数据挖掘和深度学习等领域的技术应用在食谱数据上,以便更好地开展食谱相关的研究。

在 2018 年,刘汉然<sup>[3]</sup>提出了一种食物搭配方法,该方法分析了食物及其营养成分之间的关系,基于此关系提取了营养成分的特征,然后将矩阵分解技术运用在营养成分的特征矩阵上,再对分解结果降维,从而找到食物成分之间的内部关系。该方法的提出为食物搭配研究提供了思路。同年,叶伟聪<sup>[4]</sup>结合营养学的相关知识,通过分析用户的性别、年龄、身高和体重等信息,以及分析用户对自身身体所需营养的实际需求,采用一种基于约束的食谱推荐算法,生成符合用户个人营养健康标准的食谱。

在国外,近些年也有很多关于食谱组成成分的研究。Ahn 等人<sup>[5]</sup>在他们的研究中,使用了由两个食谱网站(epicurious.com 和 allrecipes.com)提供的 56,498 条食谱,这些食谱网站是食物组合和烹饪方法的存储库,它们的大规模和多样化作为食材烹饪的基本原理和用户喜好提供了有趣的见解。这些食谱分为地理上不同的菜系,包括北美、南欧、西欧、拉丁美洲和东亚。基于这样一个大规模的食谱

数据集，他们引入了一种风味网络，用于捕捉烹饪食物之间共享的风味化合物，创新性地提出一种基于网络的方法来探索风味化合物对食物成分组合的影响，以此研究不同地区的食谱在烹饪成分上的偏好。Teng 等人<sup>[6]</sup>构建了两种类型的网络来捕获食物成分之间的关系。其中，Complement 网络用来捕获食物中的哪些成分倾向于频繁地搭配食用。Substitute 网络源自于由用户生成的修改食谱的建议，该网络可以分解成许多功能相同的食谱社区，由此可以吸引用户选择更加健康的食谱。这两种网络相结合，可以很好地为用户预测食谱等级。

国外也有一些研究者，围绕食谱展开了关于菜系分类和食谱生成方面的研究。在这些研究中，具有代表性的工作是 Naik 等人做出的一项研究。他们的数据同样基于食谱社区 allrecipes.com 的公开数据集，收集了大约 63,000 个食谱的配料和烹饪方法。食谱包含各种信息：食材、数量、说明、烹饪时间和用户评级等。在他们的工作中，使用了一个基于贝叶斯对概率的生成模型<sup>[7]</sup>，该模型是根据收集的食谱计算的。他们从用户处获取每个输入都是由一组食物组成，其中的一个食物子集可以用来生成新的食谱。他们首先选择这些食物的一个可行子集，然后生成一系列指令来操作食谱。这种使用机器学习以新的方式处理食物和特定食谱的思想<sup>[8]</sup>，给我们的研究带来了启发。

此外，还有很多研究者在烹饪菜谱的数据处理方面做了大量的工作，对食谱的结构进行不同的处理。大多数研究食谱的工作都依赖于简单的词袋模型（BOW）表示。在研究食谱作为词的配料袋的方法中，Ahn 等人建立了一个数据驱动的风味网络，将食谱成分联系在一起。具体来讲，每个风味化合物都与包含它的成分相连，形成了一个成分-化合物二分网络，有些化合物由多种成分共享。然后将这个二分网络投影到成分空间中获得风味网络，网络的节点表示成分，如果它们共享至少一种的风味化合物，则将这些节点链接在一起。每个链接的权重代表共享风味化合物的数量，则将风味网络变成一个加权网络。在此之后，Jain 等人在 Ahn 等人研究的基础上，进一步分析了在特定文化影响之下的烹饪习惯。在这项研究中，他们分析了印度菜的食谱组成以及风味成分，以寻找其典型特征。具体来说，他们量化了食物配对模式并建立了一种烹饪进化模型，该模型可以再现食物的使用分布以及菜肴的食物搭配，以识别菜肴的可解释的统计特性。Nedovic 等人<sup>[9]</sup>使用了整个菜谱语料库以及与其一起提供的食谱标签，将食谱简化为成分清单，并以字袋的方式进行分析。他们首先将高维度的组成空间可视化，然后将其映射到不同的美食中。然后使用主题建模技术（LDA）<sup>[10]</sup>从食谱中检查了基础的食物分组，使用深层置信网络（DBN）<sup>[11]</sup>来学习食物分布的生成模型，进一步简化了新颖的成分组合。

虽然词袋模型表示法可能在许多应用都足够，但是它无法捕捉到食谱数据的

大部分内部结构。在烹饪食谱的处理方面，人们已经做出了许多的努力，所使用的模型也从简单的词袋模型<sup>[12]</sup>到复杂的语义表示。为了更好地处理食谱数据，在结构化表示方法中，Tasse 等人提出了一种厨房最小指令语言（MILK），这是一种正式语言，这种语言用于描述烹饪说明中要求的动作。在设计这种语言时，作者旨在创建一组简洁的指令，这些指令代表食谱中命令性声明所要求的操作。MILK 无法详细地识别烹饪过程中发生的每个动作，但是它可以提供食谱中每个步骤的有用信息。他们使用 MILK 来开发 CURD（卡内基梅隆大学食谱数据库），这是一个以 MILK 语言手动注释的食谱数据库。此外，在 2013 年，Mori 等人<sup>[13]</sup>尝试用流程图手动注释日本烹饪食谱的流程。流程图是有向无环图（DAG），与一般 DAG 不同的是，一个食谱流程图有一个特殊的顶点——根，用于对应最终菜品。图中顶点对应于食物、烹饪工具、烹饪操作等，它们的标签由食谱文本中的字符序列及其类型组成。DAG 的弧用来表示两个顶点之间的关系。这种人工构建语料库的方式，为后来的研究提供了思路。之后，Jernsurawong 等人<sup>[14]</sup>提出了一种名为“成分指令依赖树”的数据结构来表示食谱，另外，他们还构建了一个解析器，通过该解析器，可以将食谱映射到所提出的成分指令树结构中。这种表示方式允许对菜谱进行更精细的划分和比较，这是朝着食谱语义方向迈出的重要一步。

此外，国外还有一些关于食谱应用方面的研究。Müller 等人<sup>[15]</sup>设计了一种系统，该系统利用食谱成分描述的语言特性和标准的营养知识进行匹配，自动计算来自网络的食谱成分的营养含量。该系统可以提供一种网络服务，使在家烹饪的人更加方便快捷地获得食物准确的热量和营养信息。另外，Min 等人<sup>[16]</sup>通过综合研究食谱成分、食物图片和属性（例如菜式和套餐、主菜和甜点）等信息，实现了第一个跨模态的食谱分析。他们提出一种烹饪文化分析框架，该框架可以利用丰富的食谱信息来解释和分析不同地区的烹饪文化。

### 1.2.2 生成式对抗网络相关研究

近年来，深度生成模型因其拥有对大量未标记数据的灵活学习能力而备受关注。然而，所有生成模型的共同问题是难以设计出一个良好的、可区分的、低偏置的度量指标来评估生成模型的输出，对抗性训练机制因此出现<sup>[17]</sup>。2014 年，Goodfellow 等人<sup>[18]</sup>提出了一种新型的生成模型训练框架——生成式对抗网络（GAN），这种训练方式与编码器-解码器的生成模型训练原理并不不同。GAN 网络包括两部分，分别是生成器网络和判别器网络。GAN 的生成模型用于学习样本的分布，生成器的目标是尽可能多地生成逼真的样本以欺骗判别器。判别模型用于分类或识别，判别器的目的是将生成器生成的样本与真实的样本区分开

来。生成器基于判别器的反向传播来优化自己的参数。通过设计生成器和判别器之间的极小极大博弈游戏，GAN 在很多应用中取得了成功。该框架避免了最大似然学习的困难，能够从完全随机的原始数据中学习，是近年来将无监督学习运用在复杂分布的数据方面最有前途的框架之一。

食谱的生成与食物的搭配是一种文本生成任务，如果我们想应用 GAN 框架来生成文本，则有许多挑战需要解决。人类可以随时评估生成模型生成文本的质量，例如语句是否通顺、语义是否完整、语法是否正确以及内容是否明确表达等。然而，人工智能系统很难做到这些。GAN 在文本生成方面存在难题。GAN 在图像生成领域具有良好的生成效果，因为图片属于连续数据，可以直接更新参数对其进行优化，最终生成逼真的图片。但是文本数据是离散数据，在这种情况下，判别器无法将梯度传播回生成器，也就无法进行参数更新。

于是，人们在 GAN 模型的基础上，通过各种方法和手段，尝试解决 GAN 在处理文本等离散类问题上参数难以更新的困难。Kusne 等人提出通过 Gumbel-softmax 分布来解决参数无法更新的困难，该分布是对基于 softmax 函数参数化多项式分布的一种连续近似，他们在生成离散序列的任务中，基于具有 Gumbel-softmax 输出分布的递归神经网络来评估 GAN 的生成性能。最后，可以使用基于梯度的算法来更新生成器和判别器的参数。

之后，Zhang 等人<sup>[19]</sup>将 GAN 应用于文本生成，并提出了 TextGAN 模型，该模型以卷积神经网络（CNN）<sup>[20]</sup>作为 TextGAN 的判别器，以长短期记忆网络（LSTM）<sup>[21]</sup>作为 TextGAN 的生成器，并使用光滑近似的思想逼近生成器网络的输出，从而解决离散数据梯度无法求导的困难，并降低梯度估计值的方差。在训练生成器时，目标优化函数采用的是特征匹配的方法，而不是使用原始 GAN 模型的标准目标函数。在 2017 年，Zhang 等人<sup>[22]</sup>再次对这项研究进行进一步的补充，新的研究着重讲述对抗训练的过程，生成器依然使用 LSTM 网络，判别器使用 RNN 网络。在训练生成器时，目标优化函数依然不同于原始 GAN 网络的目标函数，而是通过一个核化的差异矩阵，来匹配真实的句子和生成器合成的句子之间的高维隐藏特征分布，这种方法能够有效减轻对抗训练中的常见的模式崩溃问题。

此外，Yu 等人<sup>[23]</sup>认为当 GAN 的生成目标是离散的序列时，GAN 的生成方式具有一定的局限性，并分析了出现这种问题的原因，其一是生成模型输出的序列是离散的，导致从判别器网络传到生成器网络的梯度更新困难；其二是判别器只能评估完整的序列，但是此时判别器的指导作用就会不明显，而如果在生成器生成序列的同时，判别器来判断生成模型的效果，如何平衡当前序列得分和完整序列得分又是一个难以解决的问题。为了解决这些问题，他们提出了 SeqGAN 序

列生成框架。该框架主要解决两个问题：生成器难以通过梯度更新，而判别器难以评估不完整序列。对于第一个问题，作者将文本序列的生成过程看作强化学习（RL）系统的序列决策过程，使用策略梯度算法<sup>[24]</sup>更新了生成器的参数，避免了生成器的梯度无法求导的问题。对于第二个问题，作者借鉴了蒙特卡罗树搜索（MCTS）的思想，将判别器给生成序列的打分作为对生成器生成效果的奖励，通过 MCTS 将奖励传递给序列生成过程的各个时刻，由此判别器可以随时评估不完整的生成序列。

Fedus 等人<sup>[25]</sup>提出了 MaskGAN 模型来完成类似于完形填空的任务。MaskGAN 模型的生成器采用 seq2seq 模型，由编码模块 Encoder 和解码模块 Decoder 组成，Encoder 用于读取被遮蔽的序列，在解码过程中，编码模块提供了对缺失部分的上下文的访问，解码模块会自动补充缺失的部分；判别器的结构和生成器一样，其编码模块的输入是生成器已经补充完整的序列以及真实情况下的序列，解码模块的输出是一个 0 到 1 之间的标量，表示每个时间步的概率。MaskGAN 通过 RL 中的玩家-评委算法来训练生成模型，通过随机梯度下降法和最大似然函数来训练判别模型，训练效果良好。

过去，基于 GAN 生成的文本大多数是短文本，在生成长文本方面存在很多挑战。2017 年，Guo 等人<sup>[26]</sup>提出 LeakGAN 模型，用来生成较长的文本序列，该模型的思路打破了传统的对抗模型。该模型的判别器会将提取的部分特征告诉生成器，生成器利用这些特征信息来指导长文本的生成。生成器使用的是一种特殊的分层强化学习结构，分为上下两层结构，上层结构命名为管理者模块，下层结构命名为工作者模块，两层模块均基于 LSTM 模型。管理者模块起到一个中介的作用，具体过程为，管理者模块负责接收判别器在每个时间步传来的指导信息，然后将这些指导信息传递给工作者模块，当前时刻的输入信息与来自管理者模块的指导信息结合起来计算得到下一个词。

结合了 RL 的 GAN 模型虽然取得了较好的生成效果，但是依然存在一些没有克服的困难。大部分的 GAN 模型在应用了 RL 的启发式算法思想之后，如上文提到的分层强化学习、MCTS 等，都会导致 GAN 模型难以训练。为了解决这些困难，Nie 等人<sup>[27]</sup>提出了 RelGAN 模型，该模型是一种用于文本生成的新架构，由三部分组成：生成器以 relational memory 来替代 LSTM，用于远程依赖性建模，这种方式加快了模型的训练速度、增强了模型的表达能力，使得在长文本上的生成效果更好；在训练 GAN 模型生成离散数据时，不再使用 RL 的启发式算法，而是利用 Gumbel-softmax 松弛技术来训练 GAN 模型；判别器利用多层词嵌入<sup>[28]</sup>表示，从多个维度全面描述词向量之间的不同，为生成器提供更多信息，使生成器的更新更加多样性。RelGAN 模型是第一个使用 Gumbel-Softmax 松弛技术成

功生成接近真实的长文本的生成模型，其文本生成的效果好于其他大部分的生成模型。

### 1.3 本文的主要研究内容

基于国内外现有的对食谱生成技术和生成式对抗网络的相关研究，在本文中，我们采用了生成式对抗网络与强化学习技术相结合的文本生成模型，来生成科学合理的食谱，并将生成结果与其他深度神经网络作对比。同时，为了满足不同人群对烹饪口味的个性化需求，我们将口味标签添加到实验数据集中，应用条件-生成式对抗网络来训练所需的模型。

在获取食谱数据集时，我们使用爬虫框架获取来自美食杰网站的食谱数据，解决了以往的食谱研究中数据集获取困难的问题。为了扩充食谱数据量，提升模型的生成效果，我们对训练数据进行数据增强处理。增强的方式是随机删去每道食谱中的一个食材。

本文的主要研究内容可以分为下面两部分：

(1) 基于生成式对抗网络的食谱生成研究：首先，进行研究的基础是获得既准确又充分的食谱数据，为了得到这样的数据，我们设计并实现了 **Scrapy** 爬虫框架，该框架基于 **Python**，功能强大，帮助我们从食谱社区网站上获取了较为全面的食谱数据，随后我们对数据进行清洗，将数据整理成我们想要的格式。然后，我们应用 **SeqGAN** 框架来实现一种新颖的食物搭配和食谱生成技术。**SeqGAN** 框架采用 **LSTM** 作为模型的生成器，采用 **CNN** 作为模型的判别器，将食谱序列作为生成器的输入，由 **LSTM** 编码为目标序列，最终由 **CNN** 计算所生成的食谱与其原始输入之间的差异。**CNN** 向 **LSTM** 提供反馈，以优化其参数，直到训练过程结束。模型训练好之后，为了评价我们的模型，我们使用双语评估替补技术作为评价指标来衡量生成的食谱与真实食谱之间的相似程度。最后，我们将模型的生成效果与 **RNN** 的生成效果作比较，验证本文所提模型的优越性。

(2) 基于条件-生成式对抗网络的食谱生成研究：由于不同的用户有不同的食物口味需求，普通 **GAN** 模型不能包含输入约束，为了将个人需求纳入食谱生成中，我们提升研究方法，尝试应用 **CGAN** 模型提供具有个性化需求的食谱。**CGAN** 模型依然采用 **LSTM** 作为生成器，采用 **CNN** 作为判别器。利用 **CGAN**，在有条件（食物口味）的监督下，我们实现了在给定口味时生成对应食谱的功能。我们对数据集进行了广泛的实验，以评估模型的生成效率，实验结果证明我们的方法可以产生更好的食谱。

## 1.4 本文的组织结构

本文的内容共包含五个章节，其中，第一章是绪论，第二章介绍了本文的研究基础和生成模型的相关技术，第三章详细介绍了基于生成式对抗网络的食谱生成方法和实验结果，第四章详细介绍了基于条件-生成式对抗网络的食谱生成技术和实验结果。各个章节的具体内容如下所示：

第一章是绪论，主要介绍了研究背景、选题意义、国内外研究现状（包括食谱生成相关研究和生成式对抗网络相关研究）、本文的研究内容和全文组织结构。

第二章介绍了本文的研究基础及相关技术，包括数据采集技术、文本处理技术以及文本数据增强技术，这些都对本文的研究提供了技术支持。

第三章介绍了基于生成式对抗网络的食谱生成技术，包括 SeqGAN 模型框架、生成器网络和判别器网络的结构、SeqGAN 算法流程、数据处理过程、模型评价指标以及实验部分。

第四章介绍了基于条件-生成式对抗网络的食谱生成技术，包括 CGAN 模型框架、算法流程以及实验部分。

第五章中总结了本文的所有研究工作与进展，并计划了下一步的工作和今后的研究内容。





## 第二章 研究基础和技术

本章主要介绍本文所提模型的研究基础和相关的技术，共分为四个小节。第一节介绍了数据的采集技术，第二节介绍了文本的处理技术，第三节介绍近些年流行的深度学习技术，第四节是本章小结。

### 2.1 数据采集技术

#### 2.1.1 网络爬虫技术

近些年，互联网技术实现了飞速发展，大数据时代随之到来，网络上的数据因此呈现出爆炸式的增长速度。在这种情况下，如何从海量数据中快速且准确地寻找到用户所需要的信息，就成了一个相当棘手的问题。而网络爬虫技术可以很好地解决这个问题，帮助用户更加精准地获取想要的信息。

网络爬虫<sup>[29]</sup>是一种能够依照人们设定的某种规则去自动获取网络上的信息的程序，是搜索引擎的主要组成，所谓自动抓取信息就是指网络爬虫可以通过网页链接地址查找所需的内容，并将查询结果返回给用户，这个过程中不需要人工的操作和干预。爬虫的处理过程一般包括三个部分，分别是采集数据、处理数据和储存数据。本文在图 2-1 说明了网络爬行实现的流程图。

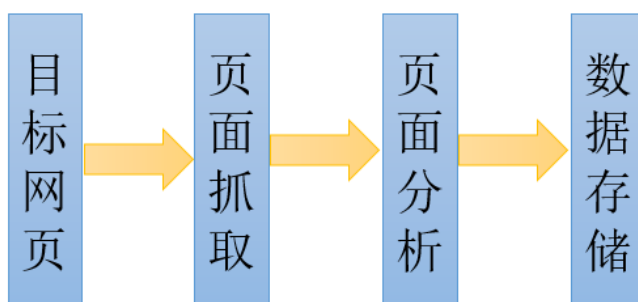


图 2-1 网络爬虫流程

网络爬虫系统框架包括三部分，分别是控制节点、爬虫节点和资源库。根据网络爬虫的系统结构及其实现技术，我们可以将其分为通用网络爬虫、深层网络爬虫、聚焦网络爬虫和增量式网络爬虫等。在实际使用网络爬虫时，通常是将多种爬虫技术结合起来使用。

其中，通用网络爬虫又被称为全网爬虫，主要用在大型搜索引擎中。该类爬虫的爬行范围广、对存储空间占用量大、爬取速度快，所以其工作性能非常强大。通用网络爬虫在爬取数据的过程中会采取某种爬行策略，常用爬行策略有深度优先爬行、广度优先爬行、反链策略和大站优先等多种爬行策略。

网络页面可以分为表层网页和深层网页，表层网页利用传统的搜索引擎即可进行索引，而深层网页隐藏在搜索表单之后，网页的信息无法像表层网页那样简单获取。深层网络爬虫<sup>[30]</sup>可以很好地对这种深层页面进行爬取。深层网络爬虫在工作过程中最重要的部分是表单填写，一种是基于领域知识的表单填写，这种方式需要提前建立一个包含领域相关内容的本体库，然后分析语义并选择关键词，最后填写表单；另一种表单填写基于网页结构分析，此方法通过将网页表示成文档对象模型（DOM）树的结构来获取表单的字段值，从而自动填写表单。

聚焦网络爬虫是一种有目的的爬虫方式，只关注与主题相关的网络内容。和通用网络爬虫相比较，这种网络爬虫增加了目的定义机制和内容过滤机制。聚焦网络爬虫能否实现，关键在于如何评价不同的网络页面内容和网络链接结构，不同的爬行策略对于页面内容和链接结构的评价结果不同，因此可以基于不同的内容和链接结构来选择合适的爬行策略。聚焦网络爬虫大致有基于网页内容评价、基于网页链接结构评价、基于增强学习和基于语境图等几种爬行策略。

当我们在爬虫过程中需要获取网站上更新的数据时，我们要用到增量式网络爬虫，这种爬虫方式只关注新产生的网页以及内容发生了变化的网页，可以保证爬虫获取的数据尽可能是新的，这样可以减轻数据的下载工作量，避免浪费算法运行时间，但缺点是算法的计算难度和实现难度会大大增加。这种爬虫方式的核心是去重，实现方式有两种，第一种是先解析页面的内容，然后根据内容判断该页面是否曾被爬虫获取过；第二种是在发送请求命令之前，判断判断需要被请求的网页地址是否曾被爬虫获取过。

这些爬虫技术都是类似的，爬虫获取数据的流程为：获取网页的链接、模拟浏览器发出请求、提取有用的数据、解析数据内容、保存获取的数据。

### 2.1.2 爬虫框架

Python 是我们现在普遍使用的一种编码语言，其内部有很多用于爬虫功能的库，例如 `urllib` 和 `urllib2` 等库，这些库主要用于抓取目标网页链接地址的内容。Scrapy 爬虫框架是基于 Python 语言的一个开源框架，能够精准有效地获取网络页面的数据信息，经常应用在数据采集、数据挖掘和信息处理等一系列场景中。Scrapy 可以在 Windows、Linux 等多种操作系统上运行，不仅提供了一些基础

组件，还提供了用户自定义功能，使得用户可以在 Scrapy 框架上自定义开发一些模块，实现网络爬虫功能<sup>[31]</sup>。

Scrapy 框架主要包括引擎、调度器、下载器、爬虫、项目管道、下载中间件和爬虫中间件等组件，其架构图如图 2-2 所示。

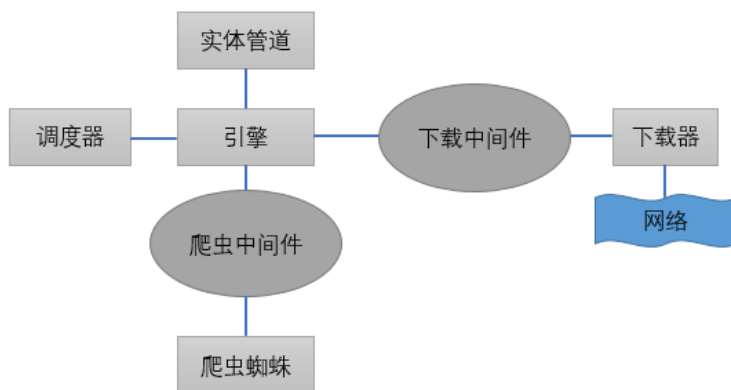


图 2-2 Scrapy 工作架构

在 Scrapy 架构中，引擎是其核心部分，也是整个框架的调度中心，相当于爬虫的“大脑”部分，主要负责控制系统内部组件间的数据流动和信号传递。

调度器用于接收从引擎发出的请求，并将这些请求加入队列中，这些请求具有优先级。当引擎再次发出请求时，调度器会将请求提供给引擎，并且会自动去除重复的网站链接，避免重复爬取相同的网址。

下载器通过引擎传来的请求获取页面数据并下载页面内容，然后将下载内容提供给引擎，引擎再将数据传递给爬虫。

爬虫是用户自定义的内容，定义了爬虫获取信息的逻辑和目标网页的解析规则等，爬虫部件在接收引擎的响应请求后，通过解析网页内容来提取数据并生成新的请求。通常，每个 Scrapy 框架中都有多个爬虫部件，每个爬虫部件专门负责处理一个或多个特定网站。

项目管道主要用于接收并处理爬虫部件从网页上提取的信息。爬虫部件解析目标页面之后，会将解析完成的数据发送到项目管道中，项目管道会按照特定顺序处理这些数据。执行过程为清洗数据、检查数据以及将数据存储到数据库中。

下载中间件位于 Scrapy 框架中的引擎和下载器之间，是一个特定的模块，主要用于处理这两部分之间的请求命令和响应。此外，下载中间件提供了一种机制，支持用户编写自定义的代码段的方法来丰富 Scrapy 的功能，这种机制非常方便灵活。

爬虫中间件位于 Scrapy 框架中的引擎和爬虫部件之间，负责处理输入到爬

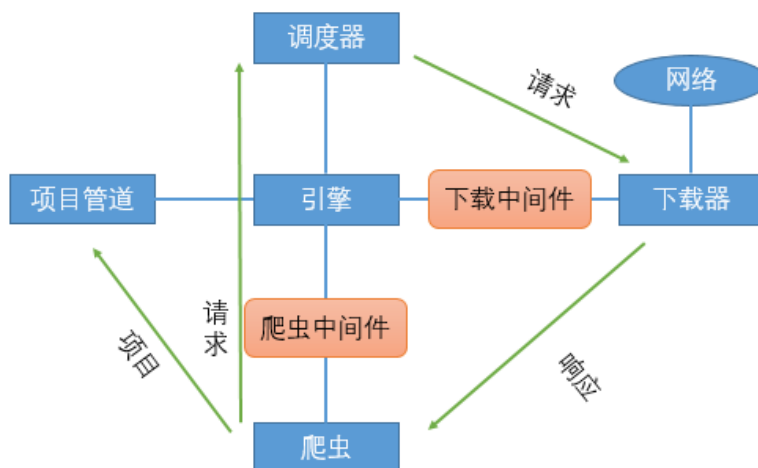
**Scrapy** 框架是所有爬虫框架中最为普遍使用的一种，其详细运行过程描述如下：

(3) 调度器接收到 Scrapy 引擎的请求命令后, 将新的请求返回至 Scrapy 引擎:

(5) 下载器下载页面信息之后，会生成一个响应，同时将下载结果返回到 Scrapy 引擎；

(7) 传回来的响应先经过爬虫部件的处理，再通过爬虫中间件返回处理后的项目以及下一步的请求，最后将这些传送给 Scrapy 引擎；

(9) 重复上述全部步骤，直到爬完所有的请求，爬虫工作结束。



## 2.2 文本处理技术

本节主要介绍文本处理方面的相关技术，包括文本预处理技术、文本表示技术和文本类数据增强等技术。

### 2.2.1 文本数据预处理

本小节介绍了文本数据的预处理过程，由于本文的研究内容围绕中国食谱展开，所以我们重点关注的是中文文本的预处理过程，即如何将中文文本经过预处理成为我们需要的数据形式。

首先是去除无用的符号。我们从互联网上获取的数据格式通常是杂乱的，在我们的有效文本之中，还存在着有一部分我们不想要的符号，这些符号会影响我们数据集的训练，因此我们需要去掉这一部分没用的符号。

其次，我们需要对中文文本进行分词操作，这一步骤是必须的。中文的分词是文本研究的基础，分词的好坏直接影响着模型的学习能力，如果中文分词良好，就可以达到自动识别语义的效果。我们将中文文本依据其语义准确地切分成多个独立的词语，而不是随机切分。现在广泛使用的分词算法主要可以分成三种，分别是基于字符串匹配的算法、基于理解的算法和基于统计的算法。

基于字符串匹配的算法又叫做机械分词算法，它按照一定的规则，将需要分词的字符串与一个足够大的词典的内容进行匹配，如果可以在这个大词典中找到该字符串，则说明匹配成功，那么该词与剩余字符串之间会被空格隔开，即分离出该词。常用的几种分词方法有以下几种<sup>[32]</sup>，一是正向最大匹配法，即从左到右依次匹配；二是逆向最大匹配法，即从右到左依次匹配；三是最少切分法，即保证每个句子中切出词的数量最少。此外，还可以将这些分词方法组合起来使用。由于汉语一个字就可以当成是词，所以正向最小匹配分词算法和逆向最小匹配分词算法很少用在中文分词中。

基于理解的算法是训练计算机以学习人类对语句的理解过程并由计算机模拟该过程，从而识别分词，并且在分词的同时分析句子的语法及其语义信息，然后利用这些信息来消除句子中存在歧义的部分。这种算法分为三部分，分别是总控系统、分词子系统以及句法语义子系统。在总控系统的控制下，句法语义子系统获得语句的语法及其语义信息，消除语句歧义，使得分词子系统可以复刻人对句子的理解，进而进行分词操作。

基于统计的算法又被称为无词典分词算法，因为这种方法不需要与切分词典的内容进行比对，只需要统计语料库中相邻字词组合起来共同出现的频率，并且

计算得到相邻字词的互现信息。互现信息是一种衡量字与字之间组合关系的紧密程度的指标，当这个值大于我们设定的阈值时，可以认为这两个字在很大概率上可以构成一个词。统计分词算法首先需要有一个统计语言模型，对句子以词为单位进行划分，然后计算划分结果的概率，保留概率最大的分词方法。主要用到的统计模型有 N 元语言模型、条件随机场和隐马尔科夫模型等。

当我们对中文分词时，句子中的词语都会被划分出来，而有些词并没有实际意义，这些词的划分会加大后续提取关键词的工作量。所以在分词以后，我们需要去除停用词以优化分词的结果，这样可以不但可以节省存储空间，还可以提高文本的检索速度。在实际操作时，我们需要从网上获取一个中文停用词表，然后根据停用词表对分词后的文本进行去除停用词操作。

### 2.2.2 文本表示技术

在很多自然语言处理任务中，文本表示是不可或缺的一步。文本表示就是将输入的文本转化为计算机可以识别并处理的向量形式，也就是抽取文本特征的过程。文本表示方法通常分为两类，即基于向量空间模型的表示方法和基于神经网络的表示方法。

向量空间模型就是将中文文本转化成由实数值组成的向量形式，每个向量的分量对应一句话中的一个词。词袋模型是一种经典的向量空间模型，它将文本字符串看成一个装满词的袋子，且里面的词无序排列，以两个词袋中重合的词的分布来判断两个词袋的相似程度。另一种计算方法是词频-逆文件频率（TF-IDF），其中，词频（TF）指某一个词语在给定文件中的出现次数，逆文件频率（IDF）用来表示某个词语的区分能力，通常来说，包含该词的文件越少，则该词的区分能力就越强，其 IDF 值越大。另一种被广泛使用的是 Word2Vec<sup>[33-34]</sup> 模型，这种模型使用 Skip-gram 和 CBOW 两种方法。其中，Skip-gram 的目标是通过当前某个词来预测该词上下文出现的概率，CBOW 则是通过上下文来预测当前单词的概率。向量空间模型的向量维度含义明确、简单直接，因此被广泛使用。

另一种向量化方法基于神经网络，这种方法应用深度学习技术对文本进行建模。最常见的神经网络是 CNN、RNN 和 LSTM 等。CNN 利用其独有的卷积层、池化层等网络结构，来提取研究对象的重要局部特征。而 LSTM 可以解决长距离的依赖问题，不会损失语义信息，擅长捕捉距离较长的信息。结合这两种网络的特点，RCNN 被提出并成功用于文本建模场景中。这些基于神经网络的文本表示方法同样被广泛使用。

### 2.2.3 数据增强技术

自然语言处理过程会受限于模型对数据的过拟合，而数据量不足很容易导致模型过拟合，使模型的泛化能力弱。因此面向文本的数据增强有助于自然语言处理任务。文本数据增强常采用以下几种方法：

(1) 反向翻译。先把句子翻译成一种不同的语言，再把句子翻译回原始的语言，然后关注新的语句与原始的语句是否相同。如果不同，则使用这个新的语句作为数据增强语句。

(2) 词汇替换。从原始的语句中随机选出一个词，然后在同义词词典中找到该词的同义词来替换选出的词。

(3) 随机插入。不考虑停用词，从句子中随机选出一个词，再在该词的同义词词典中随机选出一个词，并将该词随机插入原始句子中。

(4) 随机交换。随机打乱数据集中句子的顺序或随机交换句子中的两个词的位置。

(5) 随机删除。设定某个概率值，以这个概率随机删除标题和描述中的字或词，然后用空格代替。

## 2.3 深度学习技术

本节主要介绍一些深度学习技术，包括深度学习的概念、深度学习框架、卷积神经网络、循环神经网络、生成式对抗网络和条件-生成式对抗网络。

### 2.3.1 深度学习概念

深度学习是机器学习的一种，传统的机器学习算法受到参数数量的限制，学习特征的能力有限，无法拟合一些足够复杂的曲面，而深度学习通过建立深层次的神经网络结构来模拟人的大脑以进行分析学习，具有强大的学习表达能力，可以拟合足够复杂的特征分布。深度学习方法分为监督学习和无监督学习，擅长处理图像识别、自然语言处理、机器翻译、趋势预测和语言识别等任务。

### 2.3.2 深度学习框架介绍

深度学习的主流开发框架有 TensorFlow、Caffe 和 PyTorch 等。这些框架被广泛运用在自然语言处理、语音识别、计算机视觉和图像处理等领域。

其中，在 2008 年，一个研究机构开发出 Theano 框架，这是首个受众较广的



深度学习框架，目前已经退出了历史舞台。TensorFlow 可以看作是 Theano 的延续，因为它们有着相似的开发理念。TensorFlow 使用数据流图来进行计算，图的节点代表函数或者数学运算，边代表节点之间传递的矩阵和张量。TensorFlow 支持 Python 语言和 C 语言的运行，是现阶段使用最广泛的深度学习框架。

Caffe 是一个简单而又高效的深度学习框架，可以支持命令行、MATLAB 接口和 Python 接口，可以在 CPU 或 GPU 上运行，使用起来非常方便。

PyTorch 的设计简洁、运行速度快，以 Python 语言优先，不但可以稳定实现 GPU 加速，还支持动态图，可以很好地提升语言结构的重用性。

### 2.3.3 卷积神经网络

卷积神经网络 (CNN) 是一种新型的前馈神经网络，在近些年被广泛应用在很多场景中，主要用于提取数据的特征。CNN 的基本结构一般包括输入层、卷积层、池化层、全连接层和输出层，结构图如图 2-4 所示。

CNN 的结构具有三个特征，分别是局部连接、权值共享、时间或空间的下采样。其中，局部连接是指每个神经元只与部分神经元相连接，进而减少网络参数；权值共享是对有相同特征的数据使用同样的参数，这个特性同样可以减少网络参数；下采样即池化操作，用于降低数据空间尺寸，防止模型过拟合。

在 CNN 中，卷积层设计用于提取输入数据的局部特征，该层有一个重要结构——滤波器，滤波器相当于特征提取器，每一个滤波器都通过设定好的步长来遍历输入数据的全部内容，然后通过卷积计算抽取输入数据的特征信息。卷积层后面通常接一个池化层，该层会根据一定的规则对输入数据进行进一步的特征提取，从而大大降低数据的特征维度，避免模型出现过拟合。此外，全连接层在 CNN 的尾部，该层的所有神经元之间都有权重连接，用于计算前面提取的特征值的加权和。

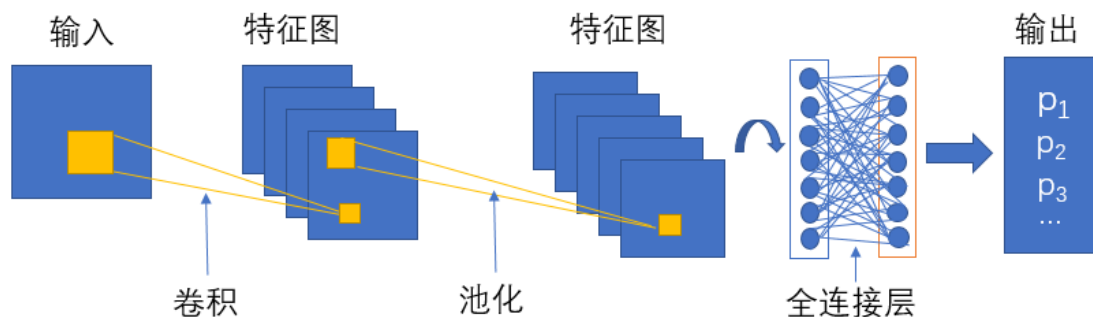


图 2-4 CNN 基本结构

起初，CNN 在图像领域取得突破，如今 CNN 在自然语言处理任务、视频预

测和语音识别等任务上也发挥着重要作用。

### 2.3.4 循环神经网络

循环神经网络 (RNN) 不同于 CNN 的前馈神经网络机制, RNN 内部存在同层节点之间的连接, 这种结构决定了 RNN 擅长于解决序列化问题, 例如语音识别和机器翻译等序列问题, RNN 可以根据上下文的状态来保存上下文信息。此外, RNN 还可以将输入数据进行定长编码, 从而保留输入数据的全局最优特征。

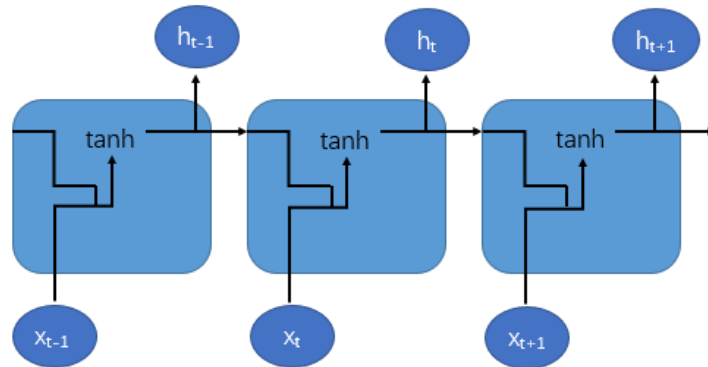


图 2-5 RNN 基本结构

图 2-5 为 RNN 的信息传递示意图, 每个 RNN 单元会接收  $t-1$  时刻的隐含层输出  $h_{t-1}$  和当前  $t$  时刻的输入  $x_t$ , 根据两者计算当前时刻的隐含层输出值  $h_t$ , 过程以此类推, 其中隐藏状态的激活函数选用  $\tanh$ 。

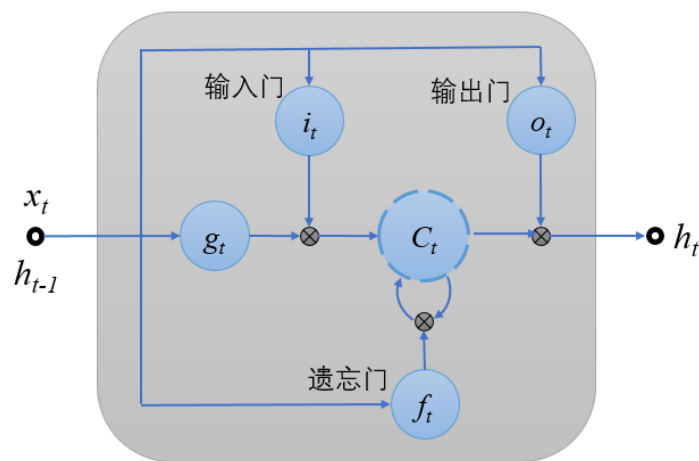


图 2-6 LSTM 门结构

从 RNN 的基本结构可以看出, 随着输入序列长度的增加, 在信息传递过程中, RNN 的隐藏层参数不断进行点乘操作, 这样容易造成信息的丢失, 为了解

决 RNN 在学习过程中的长期依赖问题<sup>[35]</sup>，长短期记忆网络（LSTM）应运而生，它记住的信息模式比 RNN 能记住的更长。LSTM 引入了一种特殊的操作块：门操作，包括输入门、遗忘门和输出门。我们可以把这些门操作理解成是信息的滤波器，其作用是获取并记住重要的信息。图 2-6 为 LSTM 的门结构示意图。具体来说，遗忘门决定了在  $t-1$  时刻的单元状态中，哪些信息可以保留到  $t$  时刻的状态  $c_t$  中，其余信息会被丢掉；而输入门决定了在  $t$  时刻的输入  $x_t$  中，哪些信息可以保留到单元状态  $c_t$  中；输出门则决定了在单元状态  $c_t$  中，哪些信息可以输出到网络当前输出值  $h_t$  中。

### 2.3.5 生成式对抗网络

2014 年，Ian Goodfellow 提出了生成式对抗网络（GAN），其对抗思想源自于博弈论中的零和博弈，对抗的双方是生成器网络和判别器网络。GAN 并不属于一种完整的网络模型，而是一个特殊的网络框架。GAN 的优点在于训练过程中仅用到了反向传播，而不需要马尔可夫链。

GAN 生成器的输入是一组随机噪声，GAN 通过学习随机噪声的分布来生成尽可能真实的数据，从而拟合出与真实数据分布尽可能一致的生成数据分布。判别器则用来判断输入的数据来自真实的数据还是生成的假数据。在训练过程中，生成器和判别器之间相互制约、对抗训练，生成器根据判别器的结果微调自己的参数，提升自己的生成能力，直到生成器可以生成判别器无法判别真假的数据，最终两个模型均达到稳态。图 2-7 为 GAN 的结构示意图。

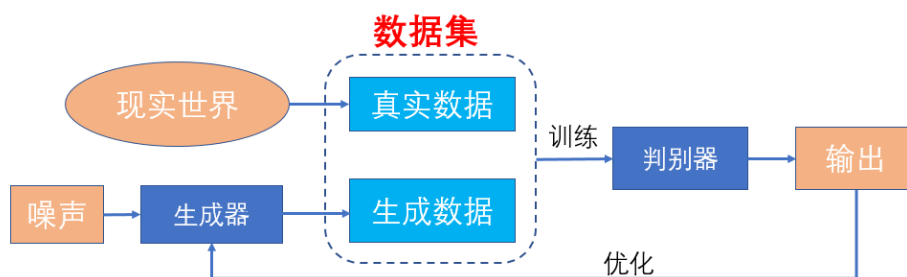


图 2-7 GAN 结构

GAN 通过训练可以学习到随机噪声的数据分布，通过其独有的对抗训练机制，生成器生成的结果通常令人满意，因此被广泛应用于数据生成领域，包括图像生成、风格转换和文本生成等任务。

### 2.3.6 条件-生成式对抗网络

GAN 的缺点是生成过程过于自由，对于较复杂的数据，GAN 的生成结果往往是不可控的，为了解决这个问题，条件-生成式对抗网络 (CGAN) 在原始 GAN 的基础上，在生成器和判别器中分别引入条件信息来指导生成器生成满足特定条件的数据，从而更加精确地生成我们期望的样本，生成器依旧要最大化地欺骗判别器。CGAN 中的条件信息的形式可以类别标签或其他模态的数据或图片等，只要这些信息可以为训练数据提供辅助性的指导功能，它就可以成为条件信息。在生成器中，先验噪声和条件信息结合起来作为网络的隐藏层表征向量，同样地，判别器网络的输入由两部分组成，一部分是真实的数据及其条件信息，另一部分是生成的数据及其条件信息，CGAN 模型在隐藏表示输入数据上具有很强的灵活性。

## 2.4 本章小结

本章主要介绍论文的研究基础及其用到的相关技术。首先介绍了基于 Python 的 Scrapy 爬虫技术，然后介绍了文本处理技术，包括文本预处理技术、文本表示技术和文本数据增强技术，最后介绍了几种主流的深度学习技术，包括深度学习概念及框架、卷积神经网络、循环神经网络、生成式对抗网络和条件-生成式对抗网络。这些技术都为本文进行的食谱生成技术研究做了很好的铺垫。



### 第三章 基于生成式对抗网络的食谱生成

在本章节，我们运用了第二章中介绍的数据处理技术和深度神经网络技术对食谱生成技术展开研究。我们在第一小节中介绍了基于 SeqGAN 的食谱生成模型框架中的生成器和判别器结构；在第二小节中介绍了模型的算法流程，包括预训练过程和对抗训练过程；在第三小节中介绍了食谱数据的获取技术以及网络模型的评价指标；第四小节中描述了我们的食谱生成实验的基本环境配置、参数设置、实验结果和结论；第五小节通过对比基于 SeqGAN 和 RNN 的食谱生成算法实验结果，证明 SeqGAN 模型的优越性；第六小节中总结了本章内容。

#### 3.1 基于生成式对抗网络的食谱生成模型框架

基于 GAN 的食谱生成模型仍然可以用图 2-7 来表示。构建食谱生成模型最有效的方式是使用长短期记忆网络作为模型的生成器。LSTM 网络是深度神经网络中最适合实现序列生成的技术，同时，LSTM 网络的不确定性很高，来自于先验分布的噪声传入 LSTM 模型的第一个单元，使句子随机生成了第一个词，而生成器网络的不确定性对于 GAN 网络的训练效果至关重要。CNN 的二进制分类模型适合作为食谱生成模型的判别器，因为它在文本分类方面优于其他网络。

LSTM 有能力单独完成食谱生成任务，但是直接将 LSTM 作为 GAN 的生成器会导致模型的梯度无法进行传播，从而无法指导文本这类离散数据的生成。在对现有工作进行整理分析后，为了解决这一问题，本章应用 SeqGAN 网络的思想来指导食谱的生成，即将强化学习的相关技术应用到 GAN 的生成器中。

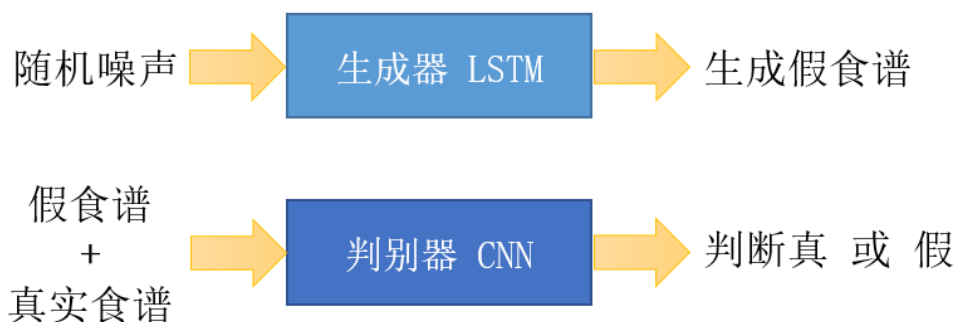


图 3-1 基于 SeqGAN 框架的食谱生成系统

SeqGAN 把生成器的输出看作一个字符选择的决策性序列，把判别器对生成数据的打分结果作为指导生成器更新的反馈信号，利用强化学习中的参数调节方法来指导生成器参数的更新。SeqGAN 的思想可以帮助我们搭建食谱生成模型，在我们的模型中，生成器使用基于 LSTM 网络的模型，我们将食谱序列作为模型的输入；判别器使用基于 CNN 的分类模型，CNN 使用多个具有不同窗口大小的卷积核来提取食谱特征，并对特征图进行最大池化操作，该模型用于计算生成器生成的食谱与真实食谱之间的相似度，为了使分类效果更好，可以向 CNN 中添加 Highway 层。激活函数选择 ReLU 函数，对于无监督模型的训练，大量实验证明 ReLU 函数的输出结果更好。最后，具有 sigmoid 激活函数的全连接层可以输出输入数据为真实数据的概率。图 3-1 为基于 SeqGAN 框架的食谱生成系统图。

## 3.2 基于生成式对抗网络的食谱生成模型算法流程

### 3.2.1 模型预训练

对于本文中使用的 SeqGAN 框架而言，在训练生成器的过程中，由于 LSTM 的参数是不确定的，几乎不会训练生成器生成效果较好的食谱文本，这种质量不好的文本被输入到判别器后，判别器就会给出很大的惩罚，此时生成器只能根据这些惩罚尽量少输出类似的食谱数据。这种参数优化方法速度非常慢，导致模型一直无法训练得到好的参数，使判别器同样没有办法正确更新，由此陷入一个恶性循环中。而预训练可以看作是参数的初始化过程，相当于在模型开始训练前，就让生成器学会判断哪种食谱数据是质量好的食谱文本。通过模型预训练，生成器网络可以在较短时间内生成质量较好的食谱文本，使生成器和判别器尽早达到均衡的状态。为了避免模型因为初始化不佳而无法收敛，所以在正式训练之前我们有一个预训练的过程。

预训练首先会随机初始化生成器网络的参数和判别器网络的参数；然后使用最大似然估计（MLE）对生成器进行预训练，以此来提升生成器训练的效率；最后我们利用生成器的输出数据来预训练判别器网络。

### 3.2.2 生成器训练

预训练生成器和判别器之后，我们需要交替训练生成器和判别器。在对抗训练的过程中，生成器和判别器都在努力优化其网络：生成器不断优化其生成逼真食谱的能力，而判别器则不断提高了区分生成的食谱和真实食谱的能力，从而形成了良性竞争，直到双方达到纳什均衡。

生成器训练流程如图 3-2 所示，训练流程依次经过了输入层、词嵌入层、LSTM 层、全连接层、输出层。下面详细介绍了各个网络层的具体作用。

- (1) 输入层：将食谱向量输入到 LSTM 中。
- (2) 词嵌入层：将上一层传入的数字索引映射为对应的词向量，所有序列均为相同长度，短序列需要在后面补 0，而长序列需要从右向左截断成固定长度。
- (3) LSTM 层：提取食谱数据的特征信息。
- (4) 全连接层：将提取的全部特征映射到样本空间。
- (5) 输出层：输出生成器生成的假食谱。

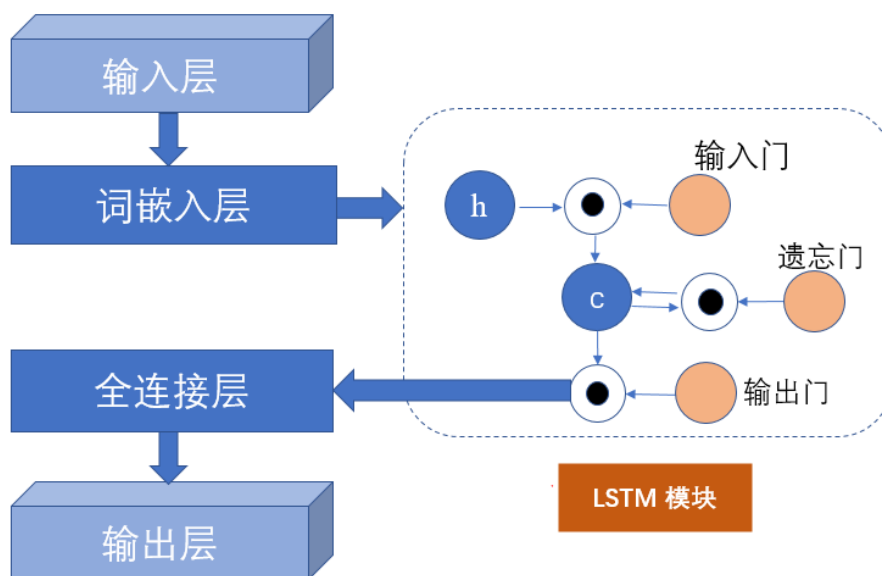


图 3-2 SeqGAN 生成器训练流程

在生成器的训练过程中，我们通过嵌入层将训练数据中的每个成分映射到一个向量，然后生成一个编码矩阵。这种表示形式有助于模型学习每种成分的含义并产生合理的结果。生成器的训练流程可以描述为，首先我们将食谱序列输入到生成器，然后我们获得每个成分的嵌入向量并将其输入到单元格中。LSTM 可以通过门结构来选择删除某些信息或将某些信息添加到当前时刻的单元状态中。LSTM 包括三种门结构，即遗忘门、输入门和输出门。遗忘门决定放弃一些信息，输入门决定更新哪些信息，而输出门则决定了模型的输出信息。最后，结合一个全连接层，我们可以得到生成器生成的假食谱。

### 3.2.3 判别器训练

在判别器的训练过程中，输入数据是两类食谱，如图 3-3 所示，一种是真实食谱，另一种是由生成器生成的假食谱。判别器的训练过程可以看作是有监督的



训练，我们将真实食谱当作正样本，并将标签设置为 1；将生成器生成的食谱当作负样本，并将标签设置为 0。

判别器的训练流程如图 3-3 所示，训练流程依次经过了输入层、词嵌入层、卷积层、激活函数 ReLU、池化层、Highway、Dropout 正则化和全连接层。下面详细介绍了各个网络层的具体作用。

(1) 输入层：将真实食谱向量和生成器生成的食谱向量输入到判别器中。

(2) 词嵌入层：将上一层传入的数字索引映射为对应的词向量。

(3) 卷积层：提取食谱数据的特征。

(4) 激活函数 ReLU：相比于 sigmoid 函数与 tanh 函数，ReLU 函数不但可以克服反向传播过程中梯度消失的问题，而且可以加快 CNN 模型的训练速度。

(5) 池化层：常见的池化方式有最大值池化、均值池化、随机池化和中值池化等，我们采用最大值池化的方式，目的是为了提取到上一层中值最大的特征，同时防止过拟合。

(6) Highway：将一部分信息处理后通过，另一部分数据不经处理直接通过，这样可以降低模型的计算复杂度以及反向传播的难度。

(7) Dropout 正则化：让网络中的某些隐藏神经元以某个概率停止工作，这样可以提高 CNN 模型的泛化能力，同时防止过拟合。

(8) 全连接层：将上一层中得到的食谱特征值进行拼接，输出样本为真实食谱的概率值。

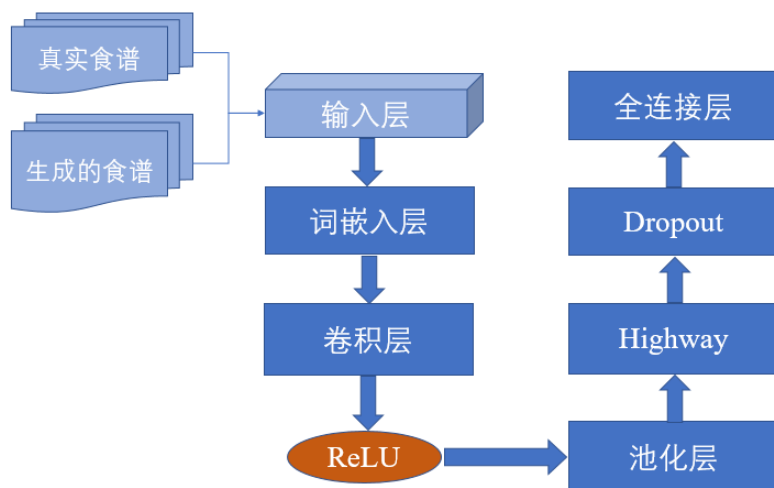


图 3-3 SeqGAN 判别器的训练流程

判别器的具体训练过程可以描述为：首先将输入序列的向量转换为矩阵，判别器采用窗口大小为 1 个词的卷积核，卷积核与输入的食谱矩阵进行卷积操作，得到食谱的特征图；然后，我们最大池化上一步得到的特征图；判别器最终输出

一个标量，代表输入数据是真实食谱的概率。训练完成后，判别器可以将真实的数据与生成的数据准确区分开。

对于生成器  $D$  和判别器  $G$  的训练是公式 (3-1) 表示的二人博弈问题。

$$\min_G \max_D V(D, G) = E_{x \sim p_{data}(x)} [\log D(x)] + E_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (3-1)$$

为了让生成器学习到训练数据  $x$  的概率分布  $p_{data}(x)$ ，我们需要为噪声数据  $z$  定义一个先验值，然后用生成器  $G(z)$  表示在噪声数据在空间上的映射，定义判别器  $D$  来输出一个标量值， $D(x)$  表示  $x$  是真实的数据而不是由生成器生成的数据，我们不断优化判别器的参数，使判别器正确判断数据来源的概率尽可能大，同时不断优化生成器使  $\log(1 - D(G(z)))$  的值尽可能小。SeqGAN 在优化过程中，生成器参数和判别器参数一直处于变化的状态。我们在实验中发现，SeqGAN 模型的生成器与判别器在训练难度上有所不同，需要采用轮数不同的交替训练方式。由于判别器比生成器更难训练，我们要给判别器设定较多的训练比例。如果生成器和判别器之间的训练无法达到平衡状态，将会使判别器过早失去对真实食谱和生成食谱的判别能力，而生成器参数也会由于判别器参数更新的失败而停止优化，最终收敛到一个生成效果不好的平衡状态。生成器与判别器之间具体的训练轮数比例需要人工通过进行多次试验来调节。具体训练过程可以描述为以下两个步骤：

(1) 首先从食谱网站上通过爬虫获得真实食谱，再从生成器中采样得到生成的食谱，有监督地训练判别器，然后更新判别器的参数。在这个过程中，生成器的参数是固定不变的。

(2) 从生成器的输出中获得生成的食谱，通过更新生成器参数来增大生成的食谱在判别器中输出为真实食谱的概率。在这个过程中，判别器的参数是固定不变的。

### 3.3 实验前的准备工作

#### 3.3.1 数据采集

为了得到本文研究所需的食谱数据，我们设计了 Scrapy 爬虫模块，该模块可以从美食网站上获得食谱的主料、辅料、口味、完成时间、烹饪步骤和功能等相关数据，并将获取的数据存在 MongoDB 中，以供后续进行实验时调用这些数据。

在爬虫过程中，首先我们在 Scrapy 框架中设置用户代理（User-Agent），用户代理可以检查用户客户端的种类及版本；然后我们在下载器中间件中开启随机切换 User-Agent 的功能。在这个过程中，我们需要维护一个包含很多用户代理的列表，每次接收到请求时，使用随机函数随机选一个用户代理传入请求头中，这样网站就很难识别出规律。

然后我们定义一个 Item 类，用于保存数据的下载地址。存储数据模块选择 MongoDB，Python 中连接该数据库的模块叫做 pymongo。为了将数据保存到 MongoDB 中，我们需要在 Scrapy 爬虫框架中开发管道，即在 settings.py 中设置启用指定的管道，管道负责将爬虫返回的数据封存在 Item 对象中。具体操作时，我们首先要在 pipeline.py 中导入 pymongo，连接程序与数据库，然后对数据库进行写操作。

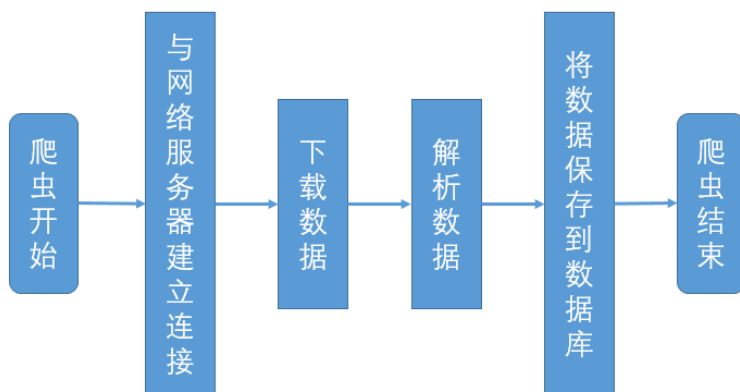


图 3-4 食谱数据获取流程

以上就是本文食谱数据的获取流程，流程图如 3-4 所示。

### 3.3.2 数据清洗

我们从网上抓取的食谱数据是不能直接被神经网络利用的，清洗数据是进行实验前的一项重要工作，是数据预处理的重要环节。数据清洗可以帮助我们删除食谱数据中的重复数据和噪声数据，保证数据的一致性。

Python 的正则表达式（re）模块为用户提供了很多正则表达式的匹配功能，适用于解析文本和分析复杂的字符串等场景。Re 模块虽然无法解决所有复杂的字符串匹配操作，但在大部分情况下都能有效实现对字符串的分析和信息提取操作。Python 可以将正则表达式转化为计算机可以解读的二进制码，然后利用匹配引擎进行深度优先遍历。

本文获取的原始中文食谱数据中含有一些特殊符号、HTML 标签、英文字母等“脏”数据，需要使用 re 库进行清洗操作。

采集和清洗食谱数据之后，我们统计本文实验所用到的食谱样本共计 10,000 条，为了扩充食谱样本数量，我们在模型训练中引入数据增强策略。

### 3.3.3 食谱成分处理

在训练 SeqGAN 模型之前，我们需要对输入 SeqGAN 模型的食谱数据进行处理，该过程主要是将食谱成分向量化，食谱成分包括主要成分和辅料成分。在进行食谱成分的向量化之前，我们需要对所有的食谱成分进行词向量的训练，然后基于训练完成的词向量索引来向量化食谱成分。

#### (1) 训练词向量

词向量，顾名思义就是用向量的形式表示词，是一种将文本中的字词进行数值化的方式，这样词与词之间就可以用定量的向量来表达关系。词向量与神经网络结合之后可以广泛地应用在自然语言处理任务中。训练词向量的方法在 2.2.2 中有介绍，我们在训练食谱成分数据时，需要考虑食谱成分之间的隐含搭配关系。

#### (2) 食谱成分向量化

文本的向量化结果应该尽可能地包含原来的空间信息，如果在向量化空间映射过程中丢失部分信息，那么该部分信息在后续的处理中也无法重新得到。

在本节中，我们基于训练完成的词向量索引，来将食谱的成分转化为向量。我们将食谱数据集中的成分与词向量进行匹配，这样食谱成分可以转化为其词向量位置上的索引编号。

此时，这些食谱成分向量的长度长短不一，因此我们需要统一指定一个向量的长度。在这里，我们计算了食谱成分的平均长度为 8，因此我们指定生成的每条食谱固定长度为 20，比 20 短的向量进行补 0 操作，比 20 长的向量对多余部分进行从右往左截断处理。至此，我们的食谱成分数据处理完成。

### 3.3.4 模型评价指标

生成模型的模型效果评估一直是一个难题，在图像生成领域和文本生成领域中都没有一个固定的评估标准。我们很难定量地表示生成的样本和真实样本之间的相似性，并且这种定量表示的结果必须与人类的观点相一致。为此，在我们的工作中采用了两种评估方法，一种是文本的相似度计算，用于定量评价生成器的生成效果；另一种是人工评估，用于规范定量评估的不足之处。下面详细地介绍这两种模型评估方法。

#### (1) 相似度计算

为了评估模型的生成效果，我们使用双语评估替代技术 (BLEU) <sup>[36]</sup> 的得分作为生成模型的评价指标，以衡量生成的食谱和真实食谱之间的相似度。两个数

据集的相似度越大，则证明生成器的效果越好。在机器翻译模型中，BLEU 用于评估翻译文本的质量，即计算机器翻译的文本和专家翻译的文本之间的相似性。尽管这个方法无法考虑到翻译文本的语法结构是否完整以及语义的是否正确，而且会忽略正确翻译的多样性，但是作为一种容易理解和操作的评估方法，在可参考的翻译文献足够多的情况下，BLEU 仍然是机器翻译领域中最有效的评估方法之一，能够达到接近于人工评价的水平。

表 3-1 BLEU 评估样例

参考文本	待评估文本
食盐，味精，白糖，调和油，葱，西红柿，青椒，鸡蛋	食盐，酱油，味精，葱，香油，柠檬，胡萝卜，鸡蛋

我们在表 3-1 中举一个简单的例子来描述本文中 BLEU 得分的计算方法。如上所示，待评估文本中共包含 8 个成分，其中，“食盐”、“味精”、“葱”以及“鸡蛋”这 4 个成分被包含在参考文本中，该待评估文本  $m$  的 BLEU 得分  $S_m$  可以表示为公式 (3-2)。

$$S_m = \frac{s}{len(m)} = \frac{4}{8} = 0.5 \quad (3-2)$$

其中， $s$  是一个累加器，用来从左到右逐词统计  $m$  中的成分在真实食谱数据中出现的次数，如果该词在真实食谱中出现，则  $s$  值加 1。生成食谱中成分的总长度用  $len(m)$  表示。

BLEU 得分的取值范围在 0 到 1 之间，用于衡量翻译文本和参考文本之间的相似度，如果这两个文本可以完全匹配，则输出的得分值是 1；如果这两个文本完全不匹配，则输出的得分值为 0；该值越接近 1 表示两个对比的文本相似度越高。

虽然这项评价技术是机器翻译领域的，但是它同样可以用在自然语言处理任务中评估生成文本的质量。本文将利用机器翻译中的 BLEU 技术作为本文的生成模型评价标准之一。在随后的实验中，我们使用“BLEU 得分”来指代本节中介绍的 BLEU 评估方法。

## (2) 人工评价

为了确保模型的评价结果更加合理，我们还提出了人工评价的方法。人工评价是指人在主观上对生成器生成的食谱文本质量的感知，人工评价的优点是人类专家对食谱质量的认知更加智能和准确，缺点是可重复性比较差、工作量大而且

评价结果容易受到评价人的主观因素的影响。我们人为设置了三个评价指标，分别是食谱完整性、无重复食材和食物可搭配食用性。每个指标的得分都是一个 0 到 1 之间的数字，得分越高代表生成的食谱在相应指标的规范下的生成效果越好。每个指标的具体含义如表 3-2 所述。

我们有三名专业人员来评估生成的食谱并按照上述三个指标为模型生成的食谱打分。在这三个指标中，食谱完整性和食物可搭配食用性这两个指标的评价依据于我们的实验数据集。

表 3-2 人工评价指标及含义

指标	含义
食谱完整性	每个食谱都包含基本的主要的食物和辅助配料
食谱无重复食材	每个食谱中不会重复出现相同的食物和配料
食物可搭配食用性	出现在同一食谱中的食物可以搭配食用

对于指标 1，我们默认大多数食谱都应该包含几种基本的食材，例如食盐、味精、葱、姜、蒜和食用油等，食谱中包含的基本食物的种类越完整，该项指标的得分越接近 1；另外，指标 2 用于反映模型的训练效果，我们不希望相同或相似的成分在同一食谱中重复出现；指标 3 用于衡量食物搭配的合理性，搭配越合理，指标 3 的得分就越高。

### 3.4 基于生成式对抗网络的食谱生成实验

#### 3.4.1 实验过程说明

本文实验所使用的主机配置和环境配置如表 3-3 所示。本文所进行的实验均在 Windows 系统下完成。考虑到实验存在较大的计算量，所以硬件配置要有 GPU，软件配置要有与 GPU 版本相适应的 CUDA、CUDNN 以及一个主流的深度学习框架。我们选择 TensorFlow<sup>[37]</sup> 框架作为实验的深度学习框架。我们使用 Anaconda 来部署 Python 环境，Anaconda 内部集成了在使用 Python 时需要的各种包及其依赖环境。

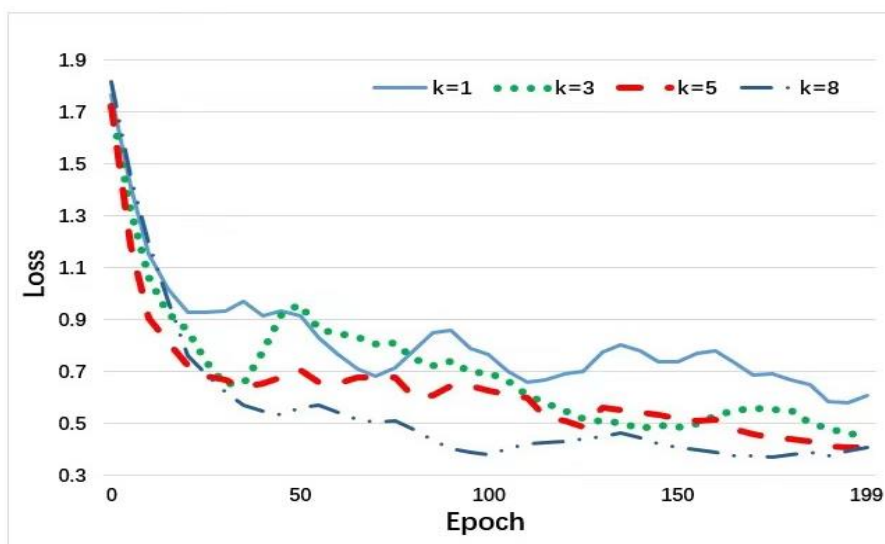
至此我们搭建好了本文实验所需的软硬件环境，接下来介绍我们的实验参数设置。根据经验，我们将模型的各项相关参数都设置为最佳值，这样可以使模型快速得到最好的生成效果。我们将 LSTM 和 CNN 存储单元的维度设置为 64 维；反向传播中的梯度截断可以将参数的更新限制在一个合适的范围内，我们将该截

断值设置为 5.0；学习速率用来表示网络中数据信息随着时间推移而累积的速度快慢，最优的学习速率取决于模型的结构和实验数据集，我们将学习速率设为 0.01。

表 3-3 运行环境配置

项目	配置
操作系统	Windows 10
中央处理器	Intel(R) Core(TM) i5-4210U
内存	16G
开发语言	Python 3.6
深度学习框架	TensorFlow 1.13.1

在 SeqGAN 的训练过程中，我们交替训练生成器和判别器，即  $k$  步优化判别器和 1 步优化生成器之间进行交替。只要生成器变化足够缓慢，判别器的结果就可以保持其最优解。超参数  $k$  的选择会影响模型收敛所需的时间以及生成器的文本生成效果。如上所述，判别器通常更加难训练，因此应逐渐增大  $k$  值。为了选择最优的  $k$  值，我们在不同的  $k$  值下分别进行了 4 次实验， $k$  分别取 1、3、5 和 8。如图 3-5 所示，我们分析了在不同  $k$  值下生成器的 Loss 值变化趋势。

图 3-5 不同  $k$  值下的 Loss 值

在实验中，我们发现在生成器和判别器在 1:1 训练比率的情况下，判别器总是弱于生成器，在该比率下，网络过早收敛并且无法完全训练；当  $k = 3$  时，模型训练不稳定。为了更准确地比较  $k$  分别取 5 和 8 时的生成效果，我们再次进行

实验, 为了保证结果公平客观, 两次实验的输入均为鸡蛋这一食材, 输出结果及 BLEU 得分情况如表 3-4 所示。

表 3-4 同一输入下不同  $k$  值的输出

$k$	输入	输出的辅料	输出的主料	BLEU 得分
5	鸡蛋	食盐, 食用油, 料酒, 胡椒粉, 糖, 香油	---	0.86
		食盐, 食用油	青椒, 西红柿	0.80
		食盐, 洋葱, 姜, 料酒, 调和油	青椒, 面条	0.75
8	鸡蛋	食盐, 调和油, 洋葱	胡萝卜, 柿子椒	0.83
		食盐, 调和油	虾, 土豆丁	0.80
		食盐, 调和油, 大蒜, 黄酒, 五香粉	柿子椒	0.62

从上表可以看出, 当  $k=8$  时, 虽然模型收敛很快, 但是 BLEU 得分不够高, 也就是说食谱的生成质量不够好, 如表 3-4 所示, 当输入鸡蛋时, 生成食谱的 BLEU 得分可以低至 0.62; 通过反复尝试, 最终将  $k$  设置为 5, 在此情况下, 当生成器训练一轮时, 判别器将训练 5 轮。

我们实验的流程主要包括食谱数据的输入、训练和输出阶段。在食谱数据输入阶段, 输入数据包括真实食谱和噪声数据; 训练阶段包括生成器的训练和判别器的训练两部分, 生成器生成的食谱和真实食谱一起输入到判别器中。在数据输出阶段, 我们设定的序列生成长度为 20。考虑到该部分实验的计算量相对较大, 我们在生成器和判别器上都采用了随机梯度下降法作为网络参数的优化算法。

本章所进行的实验均在食谱数据集上训练 200 轮, 从第 0 轮开始, 我们设置每隔 5 轮输出一次生成的食谱文本。我们设置生成模型在每次采样过程中都输出 10,000 条长度为 20 的食谱文本, 并将结果保存到 txt 文件中。

### 3.4.2 实验结果及分析

为了验证模型的食谱生成效果, 我们选择了三种常见的食物作为测试的输入, 这三种食物分别为青椒、鸡蛋、胡萝卜。对于每种输入的食物, 训练后的模型分别生成三个不同的食谱, 模型输出结果和 BLEU 得分情况如表 3-5 所示。

通过分析表 3-5 的实验结果, 我们可以清楚地观察到, 我们的模型可以生成具有几乎所有的常用辅料和具有较高 BLEU 分数的食谱。即, 该模型可以准确地学习到食谱数据集的结构, 并且可以将经常出现在同一食谱中的食物准确配对以创建新食谱。



表 3-5 SeqGAN 模型实验结果

输入	输出的辅料	输出的主料	BLEU 得分
青椒	食盐, 蒜, 酱油, 白糖	卷心菜	0.83
	食盐, 鸡精, 洋葱, 姜, 蒜, 淀粉	鸡蛋	0.75
	食盐, 调和油, 洋葱, 姜	胡萝卜, 黄瓜	0.71
鸡蛋	食盐, 食用油, 料酒, 胡椒粉, 糖, 香油	---	0.86
	食盐, 调和油	青椒, 番茄	0.80
	食盐, 洋葱, 姜, 料酒, 调和油	青椒, 面条	0.75
胡萝卜	食盐, 酱油, 洋葱, 姜, 调和油	青椒	0.88
	豆豉	柿子椒, 土豆, 五花肉, 粉丝	0.83
	色拉油, 姜, 蒜, 料酒, 调和油	黄瓜, 金橘	0.75

此外, 我们可以从实验结果中得出以下结论。

- (1) 在几乎所有食谱中, 食盐、葱、调和油和姜等都是必需的成分。
- (2) 调和油比其他食用油更常用。
- (3) 鸡蛋和青椒经常一起出现。此外, 胡萝卜和青椒经常一起出现。

以上结论也与我们的日常烹饪习惯一致, 可以认为我们写的模型的生成效果非常好。

我们将训练完成的 SeqGAN 模型嵌入到食谱生成系统中, 用户可以输入食材和想要生成食谱的数量等信息, 然后点击生成按钮, 系统会自动调用 SeqGAN 模型并生成相应的食谱。

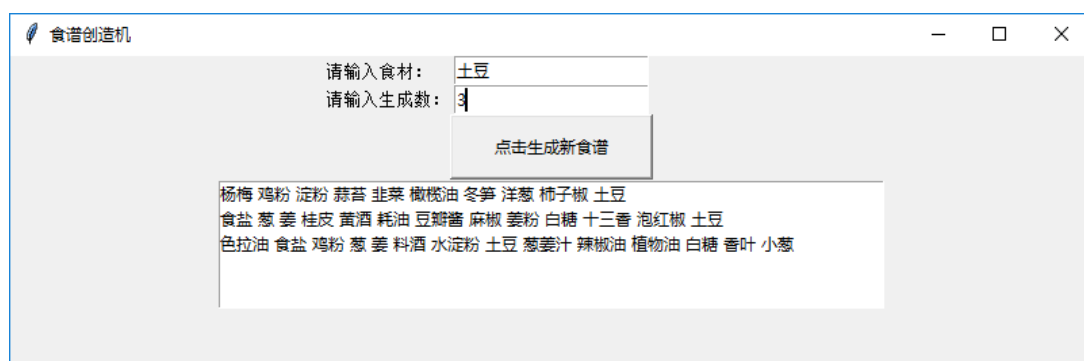


图 3-6 食谱生成系统

如图 3-6 所示, 当用户输入食材为“土豆”、生成食谱数目为“3”时, 系统会输出 3 条包含“土豆”这一食材的食谱以供用户进行烹饪选择。

### 3.5 基于生成式对抗网络和循环神经网络的食谱生成算法对比实验

为了验证 SeqGAN 模型在生成食谱质量方面的优越性,我们进行了 SeqGAN 和 RNN 的食谱生成对比试验,本节的主要内容是比较两个模型在同样输入下的食物搭配得分情况。

#### 3.5.1 基于循环神经网络的食谱生成算法

在文本生成任务中,一种传统的文本生成方法就是基于 RNN 来完成逐词生成的任务,这种根据已经生成好的序列来预测下一个词的方法是可行的。RNN 的训练基于 TextgenRNN 模型,该模型基于 Keras 和 TensorFlow 构建,并且使用注意力机制来加快训练过程并提高食谱生成质量。考虑到食谱训练集的数据量较小,且 RNN 模型计算量并不大,不需要设置过多的参数,我们的 RNN 采用两层网络结构。

#### 3.5.2 实验过程说明

本节的实验分为两个部分:第一部分是使用 RNN 网络来构建一个食谱生成模型,并在本文的食谱数据集上完成 RNN 生成效果的测试;第二部分是使用 3.4 中的 SeqGAN 框架来构建食谱生成模型,并用同样的数据集来完成 SeqGAN 模型的生成效果的测试。这两个食谱生成实验都是自由生成的,即通过输入随机噪声作为起始字符,训练模型使之可以生成指定长度的食谱序列,然后通过 3.3.4 中的模型评价指标来判断生成食谱的质量。两个实验设定生成食谱的序列长度均为 20。

对于第一个 RNN 生成实验,RNN 记忆单元的隐藏层的大小设定为 64。基于 RNN 模型的文本生成任务在训练过程中经常会遇到梯度爆炸的问题,为了避免梯度爆炸,我们将反向传播中的梯度截断值设置为 5.0。结合每次改变参数的实验结果和模型收敛速度的快慢,我们设定学习率为 0.01,设定梯度衰减系数为 0.99。本节另一个实验使用 SeqGAN 框架来构建食谱生成模型,模型设置同 3.4 中的实验。

#### 3.5.3 实验结果及分析

SeqGAN 与 RNN 模型实验结果如表 3-6 所示,表中加粗的食物为输入的食物,从表中我们可以发现,在这些食谱中,SeqGAN 模型的 BLEU 得分稳定在

0.80 左右，并且大多数得分都高于 0.80。同样，RNN 的 BLEU 分数稳定在 0.75 和 0.80 之间，某些质量不好的食谱的 BLEU 分数会低至 0.46。我们可以认为 SeqGAN 的 BLEU 分数几乎等于或高于 RNN 的 BLEU 分数。

表 3-6 SeqGAN 与 RNN 模型对比实验结果

SeqGAN 生成的食谱	BLEU 得分	人工指标得分	RNN 生成的食谱	BLEU 得分	人工指标得分
食盐，蒜，酱油，白糖， <b>青椒</b> ，白菜	0.83	0.85	食盐，蚝油，生抽，鸡精，淀粉，葱，姜，蒜，土豆，茄子， <b>青椒</b>	0.79	0.90
食盐，鸡肉精，洋葱，姜，大蒜，淀粉， <b>青椒</b> ，鸡蛋	0.75	0.80	食盐，料酒，生抽，老抽，蚝油，香油，鸡精，五香粉，酱油，辣椒，鸡精，五花肉， <b>青椒</b>	0.46	0.70
食盐，食用油，料酒，胡椒粉，糖，香油， <b>鸡蛋</b>	0.86	0.80	淀粉，食盐，酱油，味精，姜，香油，白砂糖，黄酒， <b>鸡蛋</b> ，干香菇，冬笋	0.81	0.85
食盐，调和油， <b>鸡蛋</b> ，青椒，番茄	0.80	0.60	调和油，元贞糖，鸡精，牛奶， <b>鸡蛋</b>	0.80	0.50
食盐，酱油，洋葱，姜，调和油， <b>胡萝卜</b> ，青椒	0.88	0.90	食盐，酱油，味精，葱，姜，料酒，香油，柠檬，白糖，猪肉， <b>胡萝卜</b> ，鸡蛋	0.75	0.80
<b>胡萝卜</b> ，豆豉，柿子椒，土豆，五花肉，粉丝	0.83	0.50	调和油，元贞糖，山药， <b>胡萝卜</b>	0.75	0.40

对于人工评价的指标得分，这两个模型的性能是相似。但是我们可以看到，RNN 生成的同一个食谱中会重复出现同样的食物，例如第二个食谱中的鸡精这一辅料就重复出现了。我们认为 SeqGAN 的生成性能优于 RNN，SeqGAN 生成的食谱相对完整和合理。

我们发现基于 RNN 的食谱生成模型可能出现词语重复的问题，这是因为 RNN 在训练过程中无法对完整序列信息进行评判，可以在 RNN 的基础上加入其它技术手段对生成模型进行改进和完善。此外，RNN 的运行原理是将上一时刻的输出作为当前时刻的输入，以此不断迭代进行，这就导致模型在运行过程中，如果某个时刻输出了不好的结果，那后面的部分将受到这个偏差信息的指导，最终训练出一个生成效果极差的模型。

综上所述，如果训练数据量足够大，运行环境的计算速度足够快，那么基于 SeqGAN 的文本生成模型会在序列生成任务中表现出更好的生成效果。

### 3.6 本章小结

本章主要是基于中国的食谱数据，应用适合处理食谱数据的 SeqGAN 框架，利用整理的数据对其进行训练和优化，然后进行大量实验来调整网络参数及结构，使模型生成效果最优。利用训练得到的 SeqGAN 模型，可以生成新食谱。然后将 SeqGAN 的食谱生成效果与 RNN 的食谱生成效果作比较，通过 BLEU 技术及人工评价的方法，分别为两个模型生成的食谱打分，通过比较得分，验证 SeqGAN 生成模型的优越性。



## 第四章 基于条件-生成式对抗网络的食谱生成

在本章中，我们的研究对象主要是食谱数据集及其口味标签，基于标签的文本生成是自然语言处理任务中的重要研究课题。本章基于食谱数据及其口味标签，构建了一个生成效果良好的食谱生成模型，该模型的建立可以帮助有不同口味需求的人群在日常生活中选择合适的食物以供烹饪。

模型效果的好坏与训练数据的质量密不可分，首先我们需要对实验所用的食谱数据集进行清洗过滤操作，只保留食谱的主要成分、辅料以及每条食谱的口味标签。过滤后，我们有 10,000 条质量好的食谱数据及其口味标签，标签共计 25 种口味，例如有“酱香味”、“酸辣味”、“家常味”和“蒜香味”等出现频率高的口味，以及“芥末味”、“姜汁味”和“香草味”等出现频率低的口味。

本章在第一小节中说明了食谱的口味标签对模型生成结果的影响；在第二小节中，我们设计了基于条件-生成式对抗网络的食谱生成模型的原理和框架；在第三小节中介绍了数据预处理过程和模型对抗训练的算法流程；在第四小节中描述了我们基于 CGAN 的食谱生成实验的基本环境配置、参数设置、实验结果和结论；最后在第五小节中对本章内容进行了总结。

### 4.1 条件对食谱生成结果的影响

我国地域辽阔、人口众多，来自不同地区的人们口味各异。例如，山西人在饮食中口味偏好酸味，四川、云南、贵州等地的人们偏好辛辣味，而江苏和浙江人等地的人们则更加偏好甜味。大到不同地区，小到每个家庭或者每个人对于食物都有自己的饮食口味偏好或需求。因此，我们应该为不同需求的人群提供不同的食物搭配和烹饪方法。我们需要实现，在给定一种或几种主要成分和口味的情况下，为用户提供多种食谱建议，包括烹饪主要成分和其它必要的辅料。而食物的口味主要是由辅料决定的，同一组主要成分与不同的辅料一起烹饪可以得到不同的口味。例如，当鸡蛋与醋、辣椒酱和葱一起烹饪时，它的味道是酸辣口味；当鸡蛋与食盐、葱、椒盐一起烹饪时，味道是咸鲜口味；当鸡蛋、白糖和番茄酱一起烹饪时，味道是酸甜口味。

GAN 生成器的输入是一组没有标签的噪声数据，所以其生成过程是自由的，生成结果也是不可控的，但是如果可以在 GAN 的基础上，在生成器和判别器中均引入相对应的口味标签，用于指导生成器生成指定口味下的食谱，这样就可以

精确地生成满足人们不同需求的食谱数据，引入口味标签对于我们满足个性化需求的食谱生成研究是非常必要的。

## 4.2 基于条件-生成式对抗网络的食谱生成模型框架

与 GAN 类似，条件-生成式对抗网络保留了生成器和判别器的博弈结构，不同的是，CGAN 网络在原始 GAN 中引入了条件信息，指导生成器生成指定条件下的文本或图像等。在生成器的输入端，随机噪声与约束条件结合在一起作为生成器的输入，然后由生成器生成以假乱真的食谱。判别器不仅需要识别生成的假样本和真实样本之间的相似性，还需要识别生成的样本是否满足约束条件。

通过把标签信息输入到生成器和判别器中，向模型引入了指导条件，这种基于 CGAN 的标签机制可以实现我们的研究内容。因此，我们借鉴 CGAN 的思想来设计和训练一种新的食谱生成网络模型，以满足人们日常生活中对烹饪食物选择的个性化需求。CGAN 生成的食谱比 GAN 生成的食谱更有实用价值。

在 CGAN 的生成器中，先验噪声  $z$  及其标签  $y$  以联合隐藏的形式组合在一起作为输入数据，CGAN 框架在隐藏表示方面有很大的灵活性。我们用  $x$  表示输入的数据， $y$  作为标签信息， $p_{data(x)}$  为真实数据的数据分布， $p_{z(z)}$  为随机噪声的数据分布， $G(z|y)$  为生成样本在条件  $y$  指导下的数据分布， $E$  表示平均期望。此时，生成器  $G$  和判别器  $D$  在训练过程中的目标函数可表示为公式（4-1），目标函数由前后两项组成，前项判别真实数据的损失，后项判别生成器生成数据的损失。由公式可以看出，目标损失函数的目的在于通过训练最大化判别器的区分度，而最小化生成器生成数据分布与真实数据分布之间的差异，最终使  $D(G(z|y))$  的值趋于 0.5。

$$\min_G \max_D V(D, G) = E_{x \sim p_{data(x)}} [\log D(x|y)] + E_{z \sim p_{z(z)}} [\log(1 - D(G(z|y)))] \quad (4-1)$$

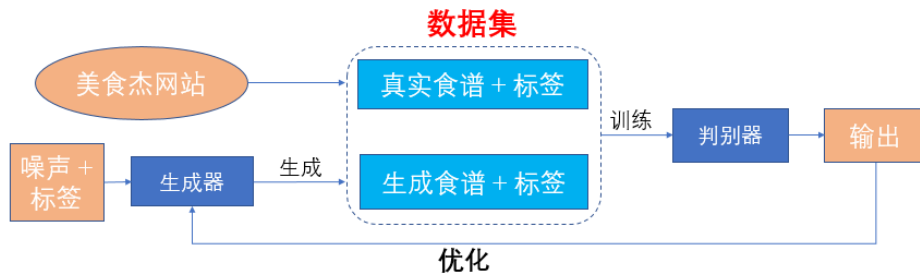


图 4-1 基于 CGAN 框架的食谱生成系统

本章中 CGAN 的基本结构如图 4-1 所示, 类似于第三章中基于 GAN 的食谱生成模型。CGAN 的生成器同样采用 LSTM 网络, 由于判别器的本质是从真实样本和生成的样本中提取和区分特征, 因此 CGAN 的判别器同样使用 CNN。

### 4.3 基于条件-生成式对抗网络的食谱生成算法

本节主要介绍了条件-生成式对抗网络的食谱生成模型的实现, 主要内容有食谱词向量的训练、对食谱数据相应的标签进行预处理、基于 LSTM 的生成器网络和基于 CNN 的判别器网络。

#### 4.3.1 食谱成分及标签处理

在训练 CGAN 模型之前, 我们同样需要预处理食谱数据, 也就是将食谱成分及其标签分别向量化表示。

##### (1) 食谱成分处理

首先处理食谱成分, 这一步骤具体操作与 3.3.3 中训练 GAN 模型时的食谱成分预处理过程是相同的。

##### (2) 食谱标签处理

在深度学习算法中, 有些样本的分类特征值不一定是连续的, 有可能是离散无序的, 而深度学习算法无法直接接收这些离散的特征值, 只能要求所有的输入和输出都是数值的形式, 所以我们需要对这些特征值进行数字化。本文中的食谱数据的口味标签正是这种离散的特征值, 这意味着我们需要把口味标签数据转换成数值的形式。对于输入到 CGAN 模型的标签数据, 我们需要预先对这些数据进行向量化。

本文清洗后的食谱数据总共包含 25 种标签, 即 25 种不同的口味, 所以, 在经过向量化之后, 每条食谱对应的标签是一个 25 维的向量。标签的向量化就是将食谱数据中的标签数据转成独热 (One-hot) 编码的向量形式。

One-hot 编码<sup>[38]</sup>, 又被称为独热编码或一位有效编码, 其原理是采用  $N$  位的状态寄存器对  $N$  种不同的状态进行编码, 其中每个状态都有其独立的寄存器位置, 而且这种编码在任何时候只有一位状态是有效的。编码没有大小之分, 每个位置上的状态都是平等的, 不同状态的数字没有比较大小的意义。One-hot 编码首先将数据的特征值映射为整数值, 然后将每个整数值都表示成一个二进制的向量, 即除整数索引被标记为 1 之外, 其它索引都是 0。

我们的口味标签共计 25 种, 故食谱的口味这一属性共有 25 个可取值, 那么口味标签就可以扩充为 25 个属性, 每个食谱的 25 个口味属性中, 只能有一个属



性的取值是 1，表示该食谱的该口味属性属于这个类别，而其余属性都用 0 来表示。

把食谱数据转换成 One-hot 编码表示有两种方式，第一种方式是使用 `sklearn.preprocessing` 中的 `LabelEncoder` 函数和 `OneHotEncoder` 函数进行数据表示形式的转换；第二种是先把食谱数据转换成 `DataFrame` 的结构，然后用 `pandas` 库中的 `pd.getdummies` 函数转换数据的表示形式。在本文中，我们使用第一种转换方式，将食谱的口味标签数据输入到 `Scikit-Learn` 提供的库函数中进行特征值的向量化操作。

#### 4.3.2 基于 LSTM 的生成器

在 CGAN 模型中，我们依然使用 LSTM 网络来提取食谱数据的特征信息，因为 LSTM 的食谱生成效果已经在第三章的实验中得到了证明，本章生成器的基本结构基于 3.2.2 中 GAN 模型的生成器。

不同的是，我们对 LSTM 结构进行了改进，在生成器网络的输入层和隐藏层之间加入了归一化层，如图 4-2 所示，归一化层用来约束网络中的参数范围，降低训练过程中的人工干预问题，加快了梯度下降过程中寻找最优解的速度，即加快了模型的收敛速度。本小节的归一化操作采用 `Layer Normalization`<sup>[39]</sup>。

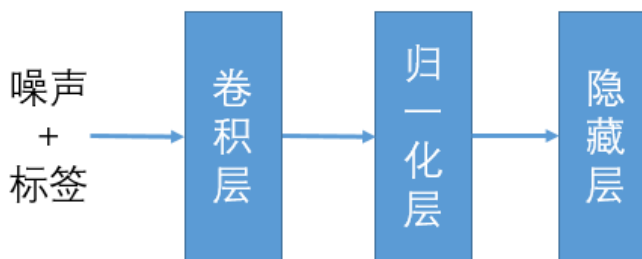


图 4-2 CGAN 生成器归一化层位置图

在 CGAN 的生成器中，噪声及其标签都映射为 LSTM 隐藏层的输入。通过对抗训练，我们可以得到一个生成效果足够好的生成器，以此生成在指定口味的条件下足够以假乱真的食谱。

#### 4.3.3 基于 CNN 的判别器

判别器的功能是通过与生成器对抗学习，不断迭代更新网络的参数，从而提高判断输入样本真假的准确率。同样地，CGAN 模型的判别器也使用 CNN 网络，CNN 对输入食谱数据的真伪性的鉴别能力也已经在第三章的实验中得到了验

证，本章判别器的基本结构基于 3.2.3 中 GAN 模型的判别器，与 GAN 判别器不同的是，我们对本章的 CNN 结构进行了一些改进，在卷积层与 ReLU 激活函数之间增加了归一化层，如图 4-3 所示，归一化操作采用 Batch Normalizaiton<sup>[40]</sup>。我们在最后一层采用一个 1x1 的卷积层来代替原来的全连接层，并使用 sigmoid 激活函数来加快模型的收敛速度。

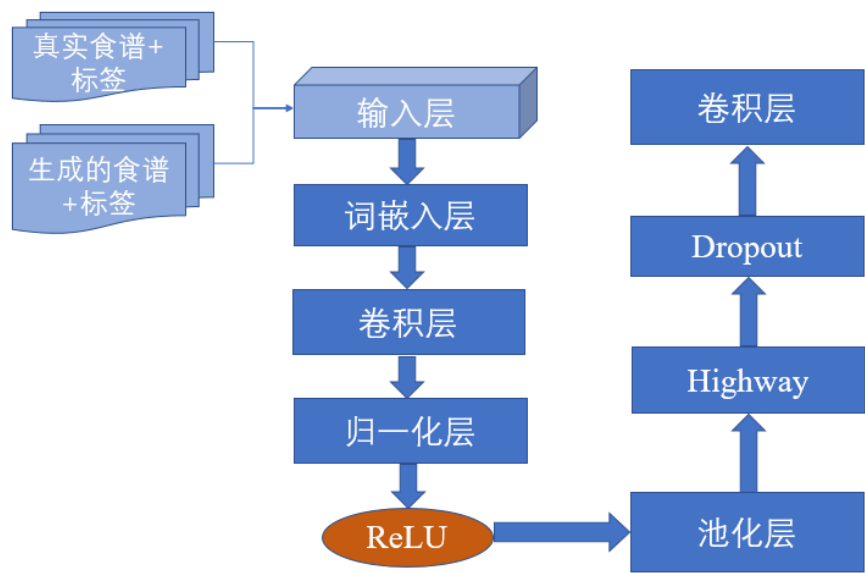


图 4-3 CGAN 判别器的训练流程

在 CGAN 的判别器中，食谱及其口味标签也将结合在一起共同作为 CNN 隐藏层的输入。通过训练，判别器可以精确地判断输入样本是否为符合口味条件的真实食谱。然后，我们将训练完成的 CGAN 模型保存在本地。通过调用此模型，我们可以输入所需的口味和成分，模型会根据标签输出满足该口味并可以与输入成分搭配食用的其他成分。

4.4 基于条件-生成式对抗网络的食谱生成实验

4.4.1 实验过程说明

本次实验依然基于 TensorFlow 这一深度学习框架，主机配置和环境配置同 3.4.1 中的表 3-2 所示。我们使用 Python 语言编写本次实验的代码，并且在第三章的食谱数据集上进行测试，以验证本文所使用的 CGAN 模型用于生成食谱任务的性能。本次实验所用的食谱数据集共计 10,000 条，标签 25 种，我们将食谱样本划分如下：训练集 8,000 条食谱，测试集 2,000 条食谱。为了实现有效的训

练,我们对数据集的训练集数据引入数据增强策略,将数据量扩充为原来的 2 倍;同时,测试集的数据规模保持不变。

实验过程如下所述:

(1) 依据 4.3.1 和 4.3.2 中食谱成分处理和标签数据处理的方法制作训练数据集的输入格式,将真实的食谱数据集与其口味标签组合在一起,真实食谱数据集作为 CNN 判别器的监督,指导 CGAN 的生成器向着我们所期望的方向生成质量好的食谱文本。

(2) 将第一步制作好的数据输入到 CGAN 中进行训练,将训练的 epoch 值设置为 200,即样本数据一共训练 200 轮;batch\_size 的值设为 64,即每轮训练时都在数据集中选择 64 条食谱数据进行训练;我们在  $k$  步优化判别器和 1 步优化生成器之间进行交替训练, $k$  的值设置为 5。模型参数设置好之后,就可以开始训练 CGAN 模型。

(3) 模型训练好后,我们将模型保存在本地,以便随后调用模型。我们采用 3.3.3 中的 BLEU 模型评价指标来评估 CGAN 的食谱生成质量。

#### 4.4.2 实验结果及分析

在本实验中,我们根据个人的口味需求评估了 CGAN 完成食谱生成任务的效率。我们分别选择了香菇、茄子和胡萝卜这三种成分作为每次实验的输入成分,对于每种输入成分,我们根据个人口味需求设置了三种不同的口味标签,实验结果如表 4-1 所示。

表 4-1 CGAN 模型实验结果

输入	口味	输出的辅料	输出的主料	BLEU 得分
香菇	咸鲜味	食盐, 味精, 姜, 料酒, 淀粉, 胡椒粉	花生, 芹菜	0.80
	香辣味	食盐, 酱油, 味精, 葱, 姜, 淀粉, 辣椒油, 猪油, 芝麻油	鸡肉	0.75
	家常味	食盐, 酱油, 味精, 姜, 白糖	柿子椒, 火腿, 冬笋, 猪大排	0.73
茄子	咸鲜味	食盐, 酱油, 葱, 蒜, 黄酒	鸡蛋	0.87
	蒜香味	食盐, 葱, 姜, 蒜, 料酒, 白糖	---	0.87
	家常味	食盐, 鸡精, 生抽	口蘑	0.67
胡萝卜	咸鲜味	食盐, 葱, 姜	干贝, 北豆腐	0.85
	香辣味	食盐, 葱, 姜, 酱油, 味精, 香油, 白糖, 辣椒粉	鸡蛋, 柿子椒	0.81
	家常味	食盐, 植物油	牛肉, 鹌鹑蛋	0.83

从表 4-1 中我们可以看出，在大多数情况下，CGAN 的 BLEU 得分都很高，大部分的 BLEU 评分可以保持在 0.75 到 0.90 之间。并且，针对每种成分，不同的口味标签指导下生成的食谱都是完整、合理且可食用的。例如，当输入的口味条件为辛辣味时，模型输出的辅料成分始终包含辣椒油或辣椒粉；当输入的口味是蒜香味时，模型输出的辅料成分包含蒜，等等。这就说明我们的口味标签是合理有效的。

我们将训练完成的 CGAN 模型嵌入到有条件指导的食谱生成系统中，用户可以输入口味、食材以及想要生成食谱的数量等信息，然后点击生成按钮，系统会自动调用 CGAN 模型并生成满足不同用户个性化需求的食谱。

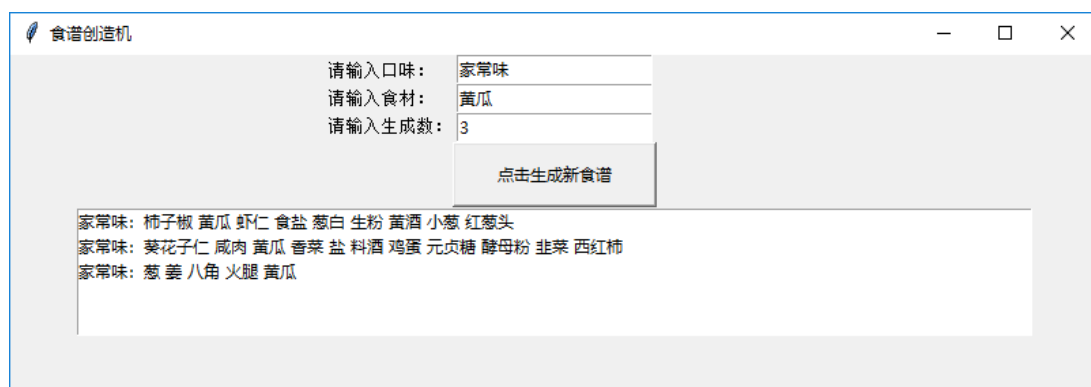


图 4-4 个性化食谱生成系统

如图 4-4 所示，当用户输入口味为“家常味”、食材为“黄瓜”、生成食谱数目为“3”时，系统会输出 3 条满足要求的食谱以供用户进行烹饪选择。

## 4.5 本章小结

本章主要介绍食谱生成任务中用于生成新食谱的条件-生成式对抗网络，主要介绍了本文 CGAN 模型的构建、生成器网络和判别器网络采用的训练方式以及 CGAN 生成实验。首先，我们介绍了口味标签对 GAN 网络的食谱生成结果的影响，即引入条件-生成式对抗网络的重要性和必要性；其次，我们介绍了 CGAN 模型的框架结构，相比于 GAN 的自由生成，CGAN 可以更好地指导食谱生成；然后，我们详细介绍了本文的食谱数据及其口味标签的处理方式，以及本文基于 TensorFlow 深度学习框架提出的 CGAN 模型中生成器网络和判别器网络的实现过程；最后，我们进行了食谱生成实验并展示出实验结果，再对实验结果进行了评价和分析，验证了本文 CGAN 模型对于食谱生成的有效性。



## 第五章 总结与展望

我们在本章总结了全文的主要研究工作以及研究进展，并且列出了下一阶段需要深入研究的内容，提出将来会从哪些方面完善我们的研究。

### 5.1 总结

近些年，生成式对抗网络由于其独特的二人博弈对抗训练机制，且仅采用反向传播、有良好的学习与生成性能，因此被广泛应用在各种研究内容中。此外，GAN 的生成器和判别器设计灵活，结合不同的技术可以应用在不同的场景中。传统的生成模型无法学习到概率密度无法计算的数据分布特征，而 GAN 网络通过对抗训练生成器网络和判别器网络，可以学习到复杂的输入数据的分布特征，不断优化生成器网络和判别器网络，最终由生成器生成以假乱真的生成样本。

随着网络上食谱社区的发展，人们越来越热衷于将自己了解的美食做法上传到网站上，和其他人共享美味的制作。日益增加的食谱数据量为本文研究食谱的生成提供了较为全面的数据集。本文正是利用这些食谱数据开展研究，首先通过爬虫技术获取了美食杰网站上的食谱数据，然后对数据进行清洗和规整。

在第三章中，我们应用了 SeqGAN 框架，在食谱数据集上训练得到 SeqGAN 模型，在这个训练过程中，我们进行了大量的实验来调整模型的参数，并将 SeqGAN 模型和 RNN 模型的生成效果进行对比。在评价模型的生成效果时，我们提出了一种人工评价的方法，与 BLEU 得分一起作为模型效果的评价指标。在第四章中，我们构建了条件-生成式对抗网络，在食谱数据集上训练得到 CGAN 模型，并使用 BLEU 技术评价 CGAN 模型的食谱生成效果。SeqGAN 模型和 CGAN 模型主要用到了 CNN 网络和 LSTM 网络。

我们第三章和第四章的实验都表明，生成式对抗网络不需要人工设计一个提取数据特征的工具，而是由网络自动学习数据的分布，这种机制适用于我们的食谱数据，具有非常优良的生成效果。

### 5.2 展望

本文下一步的研究工作计划从以下几个方面开展：

(1) 增大实验的数据集，进一步扩大食谱数据，对于深度神经网络来说，显然我们的样本数量偏少，我们需要更多的食谱数据来训练词向量，从而提升模

型的精确度和泛化能力。

(2) 改进 SeqGAN 网络和 CGAN 网络的学习能力, 我们可以考虑采用多个生成器网络或者多个判别器网络, 这样可以提高模型的泛化性能, 提高模型整体的学习能力。

(3) 尝试以食谱的其他属性作为标签来训练 CGAN 模型, 例如食谱的功能标签。功能标签有调补气血、清热去火、健脾开胃等, 以这些功能标签为 CGAN 中的指导条件, 可以使模型生成满足不同功能需求的食谱。

## 参考文献

- [1] 尹超. 社交网络节点多标签分类技术研究[D]. 成都: 电子科技大学, 2020.
- [2] 肖驰. 标签嵌入表示在文本分类任务中的应用[D]. 合肥: 中国科学技术大学, 2020.
- [3] 刘汉然. 基于成分特征提取和潜在语义分析的食物搭配推荐[D]. 淮南: 安徽理工大学, 2018.
- [4] 叶伟聪. 基于 Android 的健康食谱推荐系统研建[D]. 北京: 北京林业大学, 2018.
- [5] Ahn Y, Ahnert S E, Bagrow J P, et al. Flavor network and the principles of food pairing[J]. Scientific reports, 2011, 1: 196.
- [6] Teng C Y, Lin Y R, Adamic L A. Recipe recommendation using ingredient networks[C]//Proceedings of the 4th Annual ACM Web Science Conference. 2012: 298-307.
- [7] Thrun S. Probabilistic Algorithms in Robotics[J]. Ai Magazine, 2000, 21(4): 133-9.
- [8] Jayaraman S, Choudhury T, Kumar P. Analysis of classification models based on cuisine prediction using machine learning[C]//2017 International Conference on Smart Technologies for Smart Nation (SmartTechCon). IEEE, 2017: 1485-1490.
- [9] Nedovic V. Learning recipe ingredient space using generative probabilistic models[C]//Proceedings of Cooking with Computers Workshop (CwC). 2013, 1: 13-18.
- [10] Hofmann T. Unsupervised Learning by Probabilistic Latent Semantic Analysis[J]. Machine Learning, 2001, 42(1-2): 177-196.
- [11] Hinton G E, Osindero S, Teh Y W. A Fast Learning Algorithm for Deep Belief Nets[J]. Neural Computation, 2014, 18(7): 1527-1554.
- [12] Liu B. Text sentiment analysis based on CBOW model and deep learning in big data environment[J]. Journal of Ambient Intelligence and Humanized Computing, 2020: 1-8.
- [13] Mori S, Maeta H, Yamakata Y, et al. Flow Graph Corpus from Recipe Texts[C]//Proceedings of the Ninth International Conference on Language Resources and Evaluation. 2014: 2370-2377.
- [14] Jermurawong J, Habash N. Predicting the Structure of Cooking Recipes[C]//Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, 2015: 781-786.
- [15] Müller M, Harvey M, Elswiler D, et al. Ingredient matching to determine the nutritional properties of internet-sourced recipes[C]//2012 6th International



- Conference on Pervasive Computing Technologies for Healthcare (PervasiveHealth) and Workshops. IEEE, 2012: 73-80.
- [16]Min W, Bao B K, Mei S, et al. You are what you eat: Exploring rich recipe information for cross-region food analysis[J]. IEEE Transactions on Multimedia, 2017, 20(4):950-964.
- [17]孙博.基于生成对抗网络的文本自动生成方法研究[D].哈尔滨:哈尔滨工业大学,2018.
- [18]Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial nets[C]//Annual Conference on Neural Information Processing Systems, 2014:2672-2680.
- [19]Zhang Y, Gan Z, Carin L. Generating text via adversarial training[C]//NIPS workshop on Adversarial Training. 2016, 21.
- [20]Kalchbrenner N, Grefenstette E, Blunsom P. A Convolutional Neural Network for Modelling Sentences [C]//Proceedings of the Fifty-second Annual Meeting of the Association for Computational Linguistics,2014,1:655-665.
- [21]Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural Computation, 1997, 9(8):1735-1780.
- [22]Zhang Y, Gan Z, Fan K, et al. Adversarial feature matching for text generation[C]//International Conference on Machine Learning. PMLR, 2017: 4006-4015.
- [23]Yu L, Zhang W, Wang J, et al. Seqgan: Sequence generative adversarial nets with policy gradient[C]//Proceedings of the AAAI conference on artificial intelligence. 2017, 31(1):2852-2858.
- [24]Sutton R S, McAllester D A, Singh S P, et al. Policy gradient methods for reinforcement learning with function approximation[C]//Advances in Neural Information Processing Systems. 1999, 99: 1057-1063.
- [25]Fedus W, Goodfellow I, Dai A M. MaskGAN: Better Text Generation via Filling in the\_\_ [C]//Proceedings of the 6th International Conference on Learning Representations,2018.
- [26]Guo J, Lu S, Cai H, et al. Long text generation via adversarial training with leaked information[C]//Proceedings of the AAAI Conference on Artificial Intelligence,2018, 32(1):5141-5148.
- [27]Nie W, Narodytska N, Patel A. Relgan: Relational generative adversarial networks for text generation[C]//International conference on learning representations. 2018.
- [28]Kandola E J, Hofmann T, Poggio T, et al. A Neural Probabilistic Language Model[J]. Studies in Fuzziness and Soft Computing,2006,194:137-186.
- [29]赵文杰, 古荣龙.基于 Python 的网络爬虫技术[J].河北农机,2020(08):65-66.

- [30]孙立伟,何国辉,吴礼发.网络爬虫技术的研究[J].电脑知识与技术,2010,6(15): 4112-4115.
- [31]钱程,阳小兰,朱福喜.基于 Python 的网络爬虫技术[J].黑龙江科技信息,2016(36):273.
- [32]阳小兰,钱程,赵海廷.Web 文本预处理技术探析[J].电脑知识与技术,2010,6(29):8247-8249.
- [33]Zhang D, Xu H, Su Z, et al. Chinese comments sentiment classification based on word2vec and SVMperf[J]. Expert Systems with Applications, 2015, 42(4): 1857-1863.
- [34]Mikolov T, Sutskever I, Kai C, et al. Distributed Representations of Words and Phrases and their Compositionality[J]. Advances in neural information processing systems, 2013, 26: 3111-3119.
- [35]Liu P, Qiu X, Huang X. Recurrent Neural Network for Text Classification with Multi-Task Learning[C]//Proceedings of the 25th International Joint Conference on Artificial Intelligence,2016:2873-2879.
- [36]Papineni K, Roukos S, Ward T, et al. BLEU: a method for automatic evaluation of machine translation[C]//Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics,2002:311-318.
- [37]Abadi M, Barham P, Chen J, et al. Tensorflow: A system for large-scale machine learning[C]//12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16), 2016:265-283.
- [38]Rodriguez P, Bautista M A, Gonzalez J, et al. Beyond One-hot Encoding: lower dimensional target embedding[J]. Image and Vision Computing,2018,75:21-31.
- [39]Franceschi D D, Jang J H. Demystifying Batch Normalization: Analysis of Normalizing Layer Inputs in Neural Networks[C]//International Conference on Optimization and Learning. Springer, Cham, 2020: 49-57.
- [40]Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International conference on machine learning. PMLR, 2015: 448-456.



## 发表论文和参加科研情况

- [1] Yan Bai, Chuitian Rong and Xiangling Zhang. Food Pairing Based on Generative Adversarial Networks[C]//The 8th CCF BigData Conference,2020.（录用）
- [2] Yan Bai. A Survey on Research Progress of Generative Adversarial Networks[J]. International Journal of Engineering Research and Management, 2020, 7(10):14-15.



## 致 谢

又是一年毕业季，转眼间两年半的研究生时光匆匆而过，研究生的求学生涯即将步入尾声，初次来到天津工业大学、走进工大校门去入学报道的画面仿佛就发生在昨天。在这两年半的学习工作中，我心存感激，感激充满挑战却倍感充实的科研生活，感激我为之努力的一切，因为这些都使我成长，为我走向下一个人生阶段积累了宝贵的经验财富。

首先，我要感谢的是我的研究生导师——荣垂田老师，他不仅仅是我学习上的导师，更像是我人生道路上的引路人。荣老师毕业于名校，不仅自身知识渊博、治学严谨，更是以教书育人作为自己的工作准则。在科研工作中，他认真细致，对研究内容的细节有着极高的要求和精准的把控，他拥有高超的专业知识和技术，为我解决许多学习上的难题，使我在学业上受益颇多。在生活中，他更像是一位比我年纪稍长的朋友，给了我许多关怀，鼓励我积极向上、努力进取。是他告诉我，只要我努力，两年半的时光足够我学到很多知识和技能。从最初选择研究课题，到开展实验研究，再到现在论文的撰写，这一步步都离不开荣老师的耐心教导，我由衷感谢荣老师对我的付出和培养。衷心希望荣老师身体健康、工作顺利！

其次，我想要认真感谢我的师兄、师姐、师弟和师妹们，还有我的室友和同班同学们，非常感谢他们提供给我的帮助和指导，在这两年半时间里有他们一起，我感到非常温暖，他们总能在我身处困境时对我伸出援手，我非常珍惜与他们在学校共同学习和成长的时光，同窗之谊，终生不敢忘！

我还想要特别感谢我的家人，他们是我的依靠，是我求学路上的支撑和动力，为我倾注了无数的心血，才让我在学习和生活上有了现在的成就，我一定会继续努力，绝不辜负他们对我的期望。感谢天津工业大学为我提供了非常好的学习环境，这所学校里留下了我太多美好的回忆，我今后一定会继续努力奋斗，坚决维护它的荣誉，衷心祝愿天津工业大学可以继续走向新辉煌！

最后，感谢负责评审和答辩的各位老师，辛苦你们在百忙之中检阅我的论文，感谢你们为我提出的宝贵意见！