

学号： 2016301500030

密级： \_\_\_\_\_

# 武汉大学本科毕业论文

## 一种图像-语义互译系统

院(系)名 称： 弘毅学堂

专 业 名 称： 计算机科学与技术

学 生 姓 名： 雷伯涵

指 导 教 师： 庄越挺 教 授

黄 浩 副教授

二〇二〇年五月

# BACHELOR'S DEGREE THESIS OF WUHANUNIVERSITY

## An Image-Language Traslation System

College : Hongyi Honored School

Subject : Computer Schience and Technology

Name : Lei, Bohan

Instructor: Prof. Zhuang, Yueting

Asso Prof. Huang, Hao

MAY 2020

# 郑重声明

本人呈交的学位论文，是在导师的指导下，独立进行研究工作所取得的成果，所有数据、图片资料真实可靠。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确的方式标明。本学位论文的知识产权归属于培养单位。

本人签名: \_\_\_\_\_ 日期: \_\_\_\_\_

## 摘 要

请使用中文分号“;”分割关键词!

[illegible]

摘要内容摘要内容摘要内容摘要内容摘要内容摘要内容摘要内容摘要内容摘要内容摘要  
要内容摘要内容摘要内容摘要内容摘要内容摘要内容摘要内容摘要内容摘要内容摘要内容  
摘要内容摘要内容摘要内容

**关键词:** 关键词 1; 关键词 2; 关键词 3

# ABSTRACT

Please use English semicolon and space to separate key words.

This is abstract. This is abstract. This is abstract. This is abstract. This is abstract.  
This is abstract. This is abstract. This is abstract.

This is abstract. This is abstract. This is abstract. This is abstract. This is abstract.  
This is abstract. This is abstract. This is abstract. This is abstract. This is abstract. This  
is abstract.

**Key words:** Key1; Key2; Key3

# 目 录

1 绪论 .....	1
1.1 相关背景与需求 .....	1
1.2 技术应用意义与前景 .....	1
1.2.1 研究意义 .....	1
1.2.2 研究前景 .....	2
1.3 相关工作 .....	2
1.3.1 循环神经网络 .....	2
1.3.2 对抗生成网络 .....	3
1.3.3 自然语言处理 .....	4
1.4 本文工作目标 .....	6
1.4.1 设计大纲构思 .....	6
1.4.2 本文篇章结构 .....	6
1.5 本章小结 .....	6
2 相关技术理论 .....	7
2.1 神经网络技术 .....	7
2.1.1 循环神经网络 .....	7
2.1.2 长短期记忆网络 .....	7
2.2 GAN 生成模型选取使用 .....	8
2.2.1 基础 GAN 模型的缺陷 .....	8
2.2.2 衍生模型分类与特点 .....	9
2.2.3 StackGAN 及其衍生模型 .....	9
2.3 本章小结 .....	12
3 方案设计 .....	13
3.1 概要 .....	13
3.2 软件模型 .....	13

3.3	变量符号与定义	13
3.4	软件界面设计	14
3.5	软件功能设计	15
3.5.1	图片标注方法	15
3.5.2	自然语言生成图片方法	17
3.6	本章小结	17
4	实验与分析	18
4.1	实验环境	18
4.1.1	模型训练环境	18
4.1.2	软件样本运行环境	18
4.2	图片标注实验	18
4.2.1	模型代码	18
4.2.2	运行过程	18
4.2.3	遇到的主要问题与解决方法	18
4.3	自然语言生成图片实验	18
4.4	实验分析与总结	18
4.5	本章小结	18
5	设计意义阐述与展望	19
5.1	我的设计意义	19
5.1.1	制作了一个简洁易用系统	19
5.1.2	活用知识、实践了深度学习技术	19
5.2	未来发展方向	19
	参考文献	20
	致谢	23

附录 A 部分代码 ..... 24

    A.1 界面代码 ..... 24

    A.2 图片标注主函数代码 ..... 24

    A.3 图像生成主函数代码 ..... 24



# 1 绪论

## 1.1 相关背景与需求

随着社会发展与技术进步，我国目前越来越重视高效工具的研发。从 2000 年伊始到 2010 年互联网普及，再到 2020 年的现在，个人电子终端的功能已经越来越强大，每个人都需要更新期、更有用的效率工具。近五年，图片标注技术与生成模型都有爆炸式的发展；近两年，视觉问答技术 (Visual Question Answering)<sup>[1]</sup>、视频标注技术与秒级的图片理解更是让失明群体有了“读懂光芒”的希望，也让图片的理解从学术界或产业界的科研层面有了走向应用、走向市场的可能。

对于看不见的人来说，读懂视野这一技术是他们改善生活质量的工具，画出语言这一技术是他们表达自我的窗口；对更多的普通人群体来说，一个简洁易用的新奇效率工具，更是一种生活方式的改变，让更多的人通过机器的“魔力”，换一种角度来看语言和图片。

## 1.2 技术应用意义与前景

### 1.2.1 研究意义

图像-语义的双向翻译在目前已经有了很大的需求面。最基本的就是，图像作为视觉信号，无法被视觉失能人群感知，可以将语义转化为听觉信号，方便视觉失能人群感知世界。更进一步，对于大量的图像信号，靠肉眼处理起来需要很大的人力成本，如果使用图像标注技术，则可以从图像中提取重要信息，对语义信息进一步处理，形成简报构成参考，辅助决策。

语义的图像化更是意义非凡。基础地说，同样面对失能人群，没有视觉的人通过这一系统也有了“创作”的能力，将他们的思想从肉体的局限中有限地拓宽了出口；或者说，即使是有视觉能力的人，如果不擅长绘画，也可以通过这一技术直接表达自己思想中的画面。更进一步地说，想象力弱是很多成年人能力的局限，而一个语义向图像的转化，则可以使得一个人的表述清晰、直观地呈现在他人的视野里，可以促进交流；而表述之人也可以根据可视化的表达，发现自己表达中的问题，及时予以修正，而这是我们生活中每个人都需要的。

可以说，生成图像不仅仅是作为观赏，它可以切实际地改变我们的生活方式。

## 1.2.2 研究前景

现在并没有简单易用的商业系统，可以提供语义与图像互译的简便功能，大部分系统都只能作为技术的样品，做单向的翻译工作，目前主要在各大展览会上起到展示企业技术实力之用。对于常常需要与图片、交流打交道的人来说，这样一套系统对工作效率提高很多；广泛地，对于任何一个人，这类系统都可以改变他的生活方式。

## 1.3 相关工作

### 1.3.1 循环神经网络

#### 1.3.1.1 朴素神经网络

神经网络 (Neural Networks) 是近年比较流行的概念。在 2006 年 Hiinton<sup>[2]</sup> 提出深度学习概念后，成为了优化算法表现性能的一大利器。

神经网络的基本结构就是由神经元构成的网络，每个神经元结构如图 1.1 所示，输入输出关系如式 (1.1) 表述结果。

$$f\left(\sum_{i=1}^m w_i x_i + b\right) \quad (1.1)$$

对于第  $j$  层的每一个神经元，它会对上一层的每一个神经元输入的  $x_i$  赋予权重  $w_i$ ，计算出输出结果，即如式 (1.2) 所示，其中  $n$  是每一层的神经元个数，由输入神经元个数决定。

$$x_k^j = \sum_{i=1}^n w_{ik} x_i, k \in [1, n] \quad (1.2)$$

#### 1.3.1.2 循环神经网络

循环神经网络 (Recurrent Neural Networks, RNN) 被广泛地运用在图像处理和自然语言处理上。与一般的神经网络相同，它也有前馈层和反馈层，但是它在一般的神经网络基础上增加了基于时序的循环机制。在一般的神经网络中，神经元

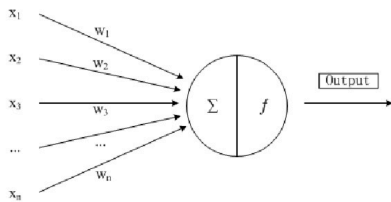


图 1.1 神经网络神经元

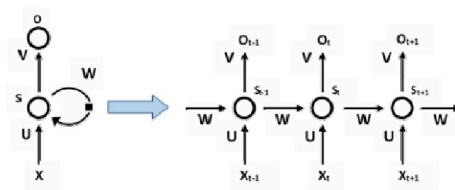


图 1.2 循环神经网络神经元

之间相互独立，但是现实中的信息往往互相关联。所以，循环神经网络的特点像人一样有记忆的能力它的输出不仅依靠当前输入，也依靠历史的输出。在一定程度上，它更能反应真实的情况。

RNN 神经元的结构与 BP 神经网络神经元有些许不同。如图 1.2所示，任一隐藏层中神经元的输出信息依靠当前时刻的输入信息和前序时序的输出信息决定，同时当前时序的输出信息也会作为下一个神经元的输入。这样的神经网络构建了神经元之间的联系，让数据有了更为有效的依赖关系，优化了信息利用的效率。

### 1.3.1.3 长短期记忆

长短期记忆 (Long Short Term Memory, LSTM) 作为循环神经网络的一个分支变种，由 Hochreiter 和 Schmidhuber 在 1997 首先提出。在循环神经网络中，一个明显的问题就是参考的上文信息距离当前神经元很远时，就难以参考，影响会变得很小，原因在于梯度消失和梯度爆炸问题。LSTM 可以保留重要信息，可以让距离比较远的重要文本对当前神经元产生比较大的影响，可以一定程度优化上下文分析的表现。

## 1.3.2 对抗生成网络

### 1.3.2.1 背景介绍

对抗生成网络 (Generative Adversarial Networks, GAN) 由 Goodfellow et al. 在 2014 年首次提出<sup>[3]</sup>。目前，它已经发展成了生成神经网络最大的热点，其研究得到了长足的发展。短短几年之内，已经有了一百余种 GAN 网络的衍生模型，其应用范围囊括了包括自然语言、图像处理、计算机视觉在内的各个领域。

在提出 GANs 模型之前，也有其他的生成式模型存在。生成方法、判别方法分别时机器学习的两大分支方向，而生成式模型则是用生成方法来生成样本的一类模型。有一类生成式模型是从人类理解角度进行设计的，比如说最大似然估计法、近似法<sup>[4, 5]</sup> 与马尔可夫链法<sup>[2, 6, 7]</sup> 等，这一类方法对于机器来说各有限制。最大似然估计法的参数更新直接受限于数据样本，数据样本不够丰富会限制生成模型的结果；近似法的目标函数太过复杂，算法只能逼近目标函数下界；马尔可夫链法的缺点便是复杂度过高。从机器理解角度设计的算法一般不直接进行拟合或者估计，而是通过采样数据样本调整模型，一般这种方法人类无法直接理解，但是生成样本是人类可以理解的。

### 1.3.2.2 基本原理

GANs 的基本结构即由一个生成模型  $G$  和一个判别模型  $D$  组成。生成模型  $G$  的目的是尽可能最小化生成与真实训练样和生成样本的区别，而判别模型  $D$  的目的则是尽可能最大化地找出真实样本和生成样本之间的区别。

通过多轮迭代生成模型  $G$  和判别模型  $D$  的对抗，可以使两个模型都达到上述的目标效果。当训练判别模型  $D$  的时候，希望输入真实样本  $x$  可以使判别器对其的判断  $D(x)$  尽量趋于 1，而生成样本  $G(x)$  通过判别器  $D$  的时候可以使得  $D(G(x))$  尽量趋于 0。在训练生成模型  $G$  的时候，输入噪声  $z$ ，希望生成的生成样本通过判别器  $D$  的时候尽量使得  $D(G(z))$  趋于 0。

可以用简单的数学变换得到公式 (1.3)，来描述训练过程。

$$\min_G \max_D V_{G,D} = \mathbb{E}_{x \sim P_{data}(x)} [\lg D(x)] + \mathbb{E}_{z \sim P_G(z)} [\lg(1 - D(G(z)))] \quad (1.3)$$

其中  $P_{data}(x)$  为真实图片集的分布。

当多轮博弈过后，极大极小问题达到最优解，即纳什均衡，当且仅当  $P_z = P_{data}$  时<sup>[3]</sup>。这时  $\mathbb{E}D(G(z))$  趋于  $\frac{1}{2}$ ，即相当于只能随机猜测 0 与 1，而生成模型  $G$  学会了真实样本的特性。

相对于传统的生成模型，我们可以发现 GANs 模型并不需要使用马尔可夫链，学习过程不需要近似推理，也不需要预先训练，自由度比较高，可以利用反向传播计算梯度，很好地利用了分段线性单元的优势，而可以回避近似计算的困难概率问题。

## 1.3.3 自然语言处理

### 1.3.3.1 背景介绍

自然语言处理 (Natural Language Processing, NLP) 是指计算机对自然语言的处理算法。所谓自然语言，就是说人类在社会文明发展过程中为了交流而逐渐发展出的语言，例如中文、英文、日文，甚至包括手语，都是自然语言。因为自然语言之间的关系远远不是函数映射那么简单的事，所以这个领域经历很长的发展历程才到了今天相对成熟的地步。

自然语言处理的发展历程大致分了三个阶段。第一个阶段是上个世纪后半叶，在二战之后洛克菲勒仅仅会的瓦伦·威佛等人在展望计算机技术的未来应用时，认为计算机可以用不同语言之间的简单词汇替换来完成翻译工作。在现在国际化知

识丰富的我们来看，显然可以看出这种方法是行不通的，但是上世纪后期，大量学者耗费人力物力，建立语言之间的词典，为每一个词汇做了映射，以实现计算机翻译。但是翻译结果并不理想，教训便是自然语言的理解需要考虑上下文关系。第二阶段是世纪之交的二十年。计算机技术迅猛发展，学者开始发现仅靠统计和替换无法完成翻译工作，而且神经网络也初步成熟，人工智能再次登上时代的焦点。例如，2005 年，Pradhan<sup>[8]</sup> 提出了语义角色解析的机器学习算法，使用 SVM 技术扩展了前人的工作<sup>[9, 10]</sup>，改进了数据的泛化功能。

第三阶段是近十年，大数据的积累帮助了自然语言处理的语料库丰富，促进了算法表现的提升；另一方面，音视频处理技术中深度学习的应用更为成熟了。其应用例如：2016 年，Y Yao et al.<sup>[11]</sup> 提出了一种基于长短期记忆的中文分词方法，突破了上下文窗口大小限制，可以保存前文的重要信息；2015 年 Jie Zhou et al.<sup>[12]</sup> 提出了不使用解析的端到端 SRL 学习系统来分析语义，使用双向 RNN，只使用文本作为输入功能，不使用语法知识。

### 1.3.3.2 自然语言应用例子：IBM Watson

2011 年，在美国的一档综艺节目中，IBM 公司云服务器 bluemix 平台上的 Watson 板块 NLP 技术相关产品大放异彩，当时因出色的对话表现爆红。

在 IBM bluemix 云平台上的 Watson 板块，有多语言对应的多种 NLP 相关应用，包括语音识别、语音生成等，经过实际试用，发现目前制作水平比较完善，但是中文包的翻译效果比较差。经过调研发现，中文的翻译训练语料库主要是使用专业文件书籍构成，导致语言不够丰富。这也给了我们一个教训，在实际 NLP 训练中，一定要使用比较丰富、贴近日常生活的语料库。

本文设计以英语为基础，使用了完整、丰富的 MSCOCO-2014 数据集。有八万多张图片作为训练集，另有四万多张图片作为测试集，且均带有相应的语义标注。这样完整的数据集更贴近一般语言环境，可以得到更好的训练效果。

### 1.3.3.3 基础依赖包

目前，常用的 NLP 算法在 python、JAVA 等语言中里面都有依赖包可以使用。

在我使用的 python 语言 3.7 版本中，可以使用依赖包 nltk(Natural Language Toolkit)<sup>[13]</sup>，这个包实现了超过五十种自然语言处理的基本算法，可以进行分词、语义角色解析等。

## 1.4 本文工作目标

### 1.4.1 设计大纲构思

本次设计需要实现如下功能，并达到要求：

1. 实现图像生成语义算法，并达到可用的效果；
2. 实现语义生成图像算法，并且图像对一般人清晰可识别；
3. 形成一个有机统一的系统，功能简洁易用。

### 1.4.2 本文篇章结构

本文共分为五个章节。

第一章，介绍了研究的背景，并通过阅读文献，整理了相关的工作，从中找出比较适合于本设计的工作，并加以总结。

第二章，详细介绍了本设计所使用技术的详情，并说明了选取技术原因。

第三章，设计了实验的方案，通过一些前人制作好的开源项目，加以修改、总结，设计系统。

第四章，详细记述了实验的结果，并通过测评手段进行测评，分析了实验结果。

第五章，阐述了设计的意义，并且展望了未来系统进一步扩展迭代的方向，指明了系统的局限性。

## 1.5 本章小结

本章介绍了设计相关领域的需求背景，也全面地介绍了这一设计的意义与前景。接下来介绍了涉及到的相关技术发展现状和相关领域的工作，也说明了其中最适合于本设计的技术及其原因。最后说明了本设计的基本要求和构思，另外设计了篇章结构，也设定了工作的目标。

## 2 相关技术理论

### 2.1 神经网络技术

本次设计中图片标注部分主要使用的技术基于长短期记忆循环神经网络。这一小节将介绍循环神经网络和长短期记忆的相关技术。

#### 2.1.1 循环神经网络

循环神经网络的结构如图 1.2所示，循环神经网络的输入比普通神经元多了一个  $w$ ，也就是上一时序的自我输出。每一个神经元的计算方式可以表述为式 (2.1)。其中  $x_t$  代表  $t$  时刻神经元的输入， $s_t$  代表  $t$  时刻神经元的状态， $o_t$  代表  $t$  时刻神经元的输出。

$$\begin{aligned} s_t &= f(u \cdot x_t + w \cdot s_{t-1}), \\ o_t &= g(s_t \cdot v). \end{aligned} \quad (2.1)$$

#### 2.1.2 长短期记忆网络

从主体上看，LSTM 与 RNN 类似，都是循环神经网络这样的链式结构。而在神经元上看，LSTM 细胞节点要比普通的 RNN 节点要复杂一些。LSTM 改进后的神经元由四个不同的神经网络层进行信息的交互，如图 2.1所示。

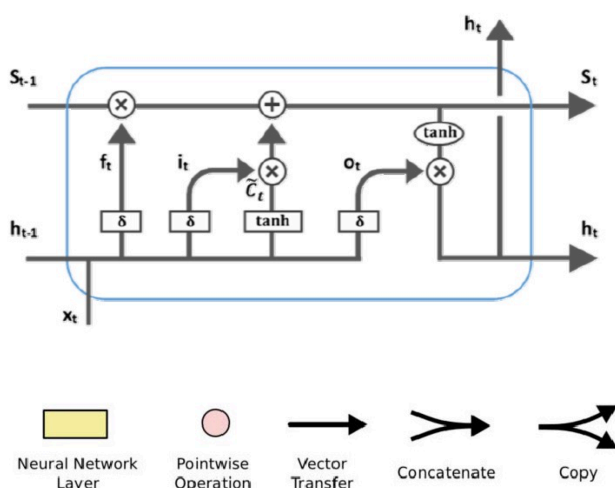


图 2.1 LSTM 神经元细胞结构图

其中圆形代表基本操作，矩形代表学习得到的状态，箭头代表数据的传输，而箭头的分裂与合并则代表向量的拆分与连接； $s$  代表状态， $x$  代表输入， $h$  代表输出。

在图 2.1 中的上方长线段代表了神经网络的状态  $s$ ，它在每个时刻的操作仅在两个地方与输入数据有交互，这就保证了  $s$  的变化不大，神经元状态比较稳定。LSTM 神经元的数据更新由“门”来进行，三个门分别是输入门、遗忘门和输出门，LSTM 神经元由此实现数据的更新和储存。

门上的行为由一个点乘 Pointwise 操作和一个激活函数 *Sigmoid* 所组成，该函数取值 0 至 1，分别代表着允许部分信息通过——其中 0 代表不允许任何信息通过，1 代表着允许所有信息通过。具体的门信息传递机制由以下几个部分完成：

1. 遗忘门：决定历史信息的保留与遗忘。通过 *Sigmoid* 函数来决定状态  $S$  的遗忘程度。

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \quad (2.2)$$

2. 输入门：决定存储到细胞状态内的新信息。先用 *Sigmoid* 函数决定记忆的更新程度，然后决定更新的内容，最后将更新的内容加在遗忘内容之上，形成当前时刻的状态。

$$\begin{aligned} i_t &= \sigma(W_i[h_{t-1}, x_t] + b_i) \\ \tilde{C}_t &= \tanh(W_c[h_{t-1}, x_t] + b_c) \\ S_t &= f_t \cdot S_{t-1} + i_t \cdot \tilde{C}_t \end{aligned} \quad (2.3)$$

3. 输出门：决定输出的值。首先用 *Sigmoid* 函数决定从状态中输出的部分，然后用  $\tanh$  对细胞状态进行处理，并与 *Sigmoid* 函数值相乘，实现输出想输出的部分。

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(S_t) \end{aligned} \quad (2.4)$$

通过这三个门对神经元的共同作用，LSTM 可以完成神经元的基本状态变换，从而完成较好的训练效果。

## 2.2 GAN 生成模型选取使用

### 2.2.1 基础 GAN 模型的缺陷

GANs 模型的缺点对应着优点，比较明显。由于 GANs 模型自由度太高，在面对过于清晰的图片等训练样本时，收敛性表现较差，生成模型可能出现退化，重复生成相同样本点，导致判别器无法工作，进而导致模型崩溃。因此，在训练过程中，调整好两个模型网络的平衡与同步非常重要。



正因为 GAN 模型有着这诸多的缺点，所以要选取比较合适的 GAN 模型来生成图片。

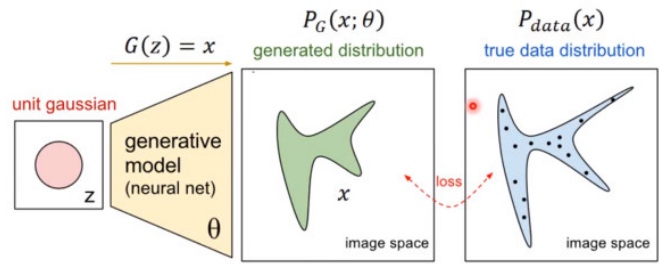


图 2.2 一般 GAN 网络模型结构

### 2.2.2 衍生模型分类与特点

目前对抗生成网络的衍生模型众多，其优化方式大抵由两个大方向衍生而来。一个方向是由损失函数的改变来优化 GAN 模型的效果，另一个方向则是从模型使用的角度来优化 GAN 模型的效果。在表 2.1 中，列举了一些最为常见的 GAN 模型衍生模型。

表 2.1

从损失函数角度 提出的优化 GAN 模型 <sup>[14]</sup>	从模型应用角度提出的优化 GAN 模型		
	网络构架角度 <sup>[15]</sup>	编码器角度	其他角度改进
Least Square GANs, Loss-Sensitive GAN, Fisher GAN, WGAN, WGAN-GP, WGAN-LP, f-GANs <sup>□</sup> DRAGAN 等	CGAN, DCGAN, InfoGAN, StackGAN <sup>[16]</sup> , AL-GAN 等	BEGAN, VAE-GAN, tDCGAN, BiGAN, 文献中 的算法 <sup>[17-19]</sup> 等	LAPGAN, ESRGAN, SRGAN, 3D-GAN, MGAN 等

图像生成的模型与基于网络构架优化的 GAN 网络模型最为贴合，本文设计使用 StackGAN 作为基础，并针对相关实现进行优化。

### 2.2.3 StackGAN 及其衍生模型

Han Zhang et al. 在 2017 年提出了 StackGAN 模型，创新性地提出了基于栈的对抗式生成网络模型从文本生成图片的方法。

StackGAN 和 CGAN 同属于优化了 GAN 网络的构架结构的一类 GAN 模型的分支。其中，StackGAN 是对相对朴素的 CGAN 模型的优化。

### 2.2.3.1 从 CGAN 到 StackGAN

条件对抗生成网络 (Conditional GAN, CGAN) 是 Mirza<sup>[15]</sup> 于 2014 年提出的, 如公式 (2.5) 表述, 对生成模型和判别模型输入文本信息作为条件, 使用了一个隐藏层的全联通网络 (MLP) 结构, 实现了通过文字标签生成图像的功能。

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim P_{data}(x)} [\lg D(x_i | y)] + \mathbb{E}_{x \sim P_G(z)} \lg [(1 - D(G(z_i | y)))] \quad (2.5)$$

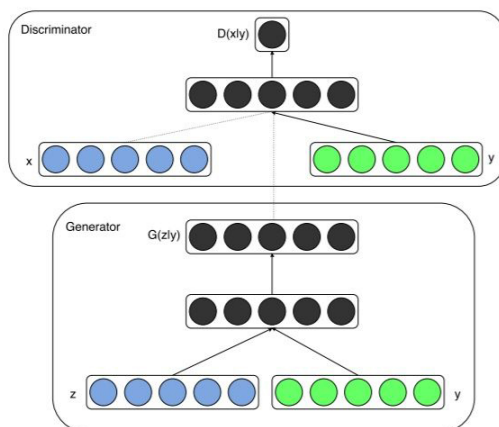


图 2.3 CGAN 构架示意<sup>[15]</sup>

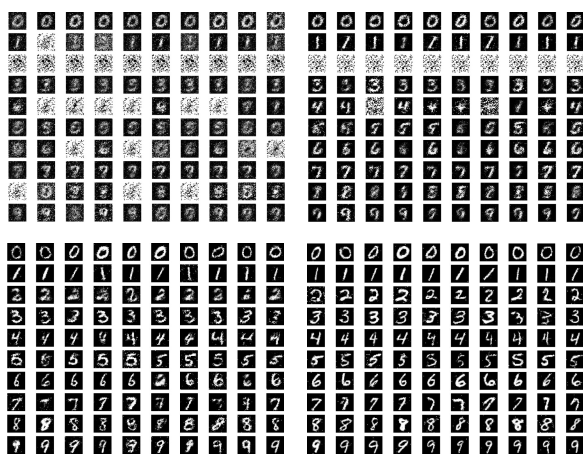


图 2.4 CGAN 生成手写数字<sup>[15]</sup> (分别经历 1、10、100、1000 个 epoch 后的结果)



图 2.5 CGAN 生成二次元头像<sup>[20]</sup>

但是, CGAN 的模型结构简单, 效果差, 收敛速度慢。在 CGAN 中, 我们将文本作为条件输入生成器和判别器来训练两个模型生成和判别图像。如图 2.4 和图 2.5 所示, 及时 CGAN 应用广且有趣, 但图像较小, 质量不高。最基础的 CGAN 算法只能生成  $64 \times 64$  像素的图片, 经过后来的改进, 也只能将图片质量提升到

128 × 128 像素。

为了解决这一问题，Han Zhang<sup>[16]</sup> 在 2016 年提出了 StackGAN 模型，使用基于栈的结构，分两步进行训练。

如图 2.6所示，第一步先训练模型使模型生成 64 × 64 像素的图片，训练出的模型只能输出像素低、形状有所扭曲的图片。第一次训练的时候并不使用转化为向量的文字直接作为输入，因为这一向量为 1024 维，过于稀疏，不利于深度的训练。在经历一次 FC 以后进行输入，这样可以增强表现。第一阶段训练的损失函数为式 (2.6) 所示。第二次训练并不增加噪声输入，经过多轮残差网络迭代，模型修正一轮训练生成图片的错误，并且在图像中添加细节，生成 256 × 256 像素的“高清图”。第二阶段训练的损失函数如式 (2.7) 所示。

$$\begin{aligned}\mathcal{L}_{D_0} = & \mathbb{E}_{(I_0, t) \sim P_{data}} [\lg D_0(I_0, \varphi_t)] \\ & + \mathbb{E}_{z \sim P_z, t \sim P_{data}} [\lg(1 - D_0(G_0(z, \hat{c}_0), \varphi_t))],\end{aligned}\quad (2.6)$$

$$\begin{aligned}\mathcal{L}_{G_0} = & \mathbb{E}_{z \sim P_z, t \sim P_{data}} [\lg(1 - D_0(G_0(z, \hat{c}_0), \varphi_t))] \\ & + \lambda D_{KL}(\mathcal{N}(\mu_0(\varphi_t), \Sigma_0(\varphi_t)) \parallel \mathcal{N}(0, I)) \\ \mathcal{L}_D = & \mathbb{E}_{(I, t) \sim P_{data}} [\lg D(I, \varphi_t)] \\ & + \mathbb{E}_{s_0 \sim P_{G_0}, t \sim P_{data}} [\lg(1 - D(G(s_0, \hat{c}), \varphi_t))],\end{aligned}\quad (2.7)$$

$$\begin{aligned}\mathcal{L}_G = & \mathbb{E}_{s_0 \sim P_{G_0}, t \sim P_{data}} [\lg(1 - D(G(s_0, \hat{c}), \varphi_t))] \\ & + \lambda D_{KL}(\mathcal{N}(\mu(\varphi_t), \Sigma(\varphi_t)) \parallel \mathcal{N}(0, I))\end{aligned}$$

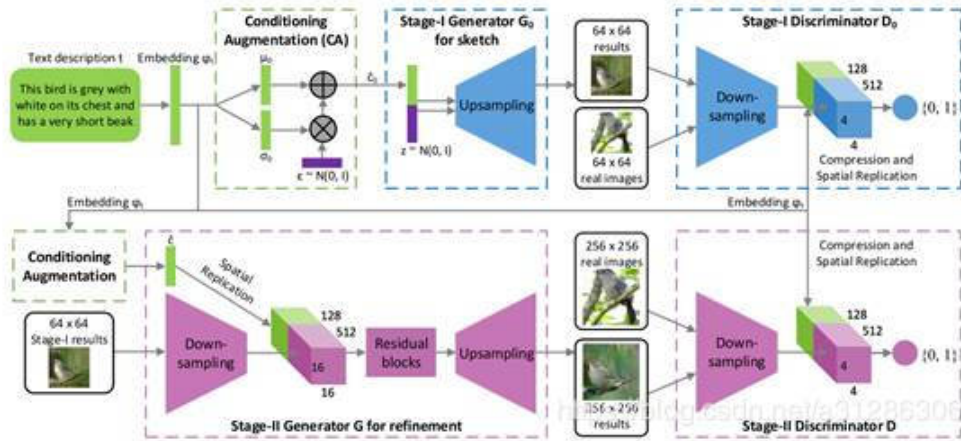


图 2.6 StackGAN 构架示意

StackGAN 模型相比较于 CGAN 模型，可以生成更高清的图片，而且结构更为真实。

### 2.2.3.2 对 StackGAN 的进一步优化与运用

同一课题组紧接着提出了 StackGAN++<sup>[21]</sup>，对原有算法进一步进行了优化。StackGAN++ 模型也叫做 StackGAN\_v2，对之前的 StackGAN 模型主要有三点改进。第一，改为采取树状结构作为输入，通过不同的生成器来生成不同尺寸的图片，多层次叠加形成了有位置关系的“假图片”，即混合图层叠加图片；第二，额外引入了非条件性的损失，直接将服从正态分布的损失  $z$  叠加在最后生成的“假图片”上；第三，引入了色彩限制，限制了最终“假图片”的色彩信息分布。

李飞飞课题组的 Johnson<sup>[22]</sup> 在 StackGAN 模型之上，提出了自己的优化。

该方法主要有三个特点：第一是有一个处理场景图的方法，第二是确保了生成的图像中物体正确，且位置关系合理。第三是确保了生成图像的质量比较好。其具体技术路线将在下一章进行推导、介绍。

## 2.3 本章小结

本章中我主要总结了本次技术方案的前置技术，包括神经网络中的 LSTM-RNN 长短期记忆循环神经网络模型和 StackGAN 模型及其相关技术。LSTM-RNN 模型是一种相对稳定、收敛，可以一定程度上避免梯度消失的深度学习模型，主要为下一章实现图片标注方法的技术做好铺垫。StackGAN 及其升级版本是 Han Zhang 及其他科研人员 2016 年其后提出的一系列理论，我简要介绍了它的背景、原理和与后续技术的比较，决定使用李飞飞课题组的场景图像生成方法作为实现的理论依据，也为下一章自然语言生成图片的方法做好了前置说明。

## 3 方案设计

### 3.1 概要

设计 requirements 是：用户输入一张图片后，可以得到相应的标注语句，了解图片语义；在用户输入一句话的时候，可以得到描述这一句话的图片，直观“感受”文字。

软件设计主界面使用 `python` 的 `tkinter` 包来制作。`python` 是一个跨平台语言，用 `python` 制作方便产品在不同平台使用。界面中应当可以选择调动两个功能，分别是图片翻译为自然语言和语句翻译为图片。两个功能分别提前训练出成品模型，通过主界面按钮内嵌入的方法，调用对应模型进行计算。

### 3.2 软件模型

软件分为三个部分，第一部分是软件界面，需要实现简洁明了的操作功能，方便使用。

第二部分是图片标注功能。实现这一功能的模型通过 `image caption` 的模型进行训练，算法参照蒙特利尔大学 Kevin Xu<sup>[23]</sup> 提出的模型进行实现，并经过基于一个或多个数据集进行多个 `epoch` 的训练，比较选出表现较好的模型，嵌入到应用中使用。

第三部分是文本生成图片功能。实现这一功能的模型通过一个特殊的 `GAN` 模型进行训练，并且要加入后期图片处理算法，以骗过判别器并使其更加真实。这一模型参照卡耐基-梅隆大学 Justin Johnson<sup>[22]</sup> 提出的基于场景双 `GAN` 模型配合进行实现，并经过合适的数据集训练，得到表现较好的模型，嵌入到应用当中使用。

### 3.3 变量符号与定义

文中使用集合如下：

1.  $O$  代表物品 (object) 的集合；
2.  $C$  代表物品目录 (category) 的集合；
3.  $R$  代表关系 (relationship) 的集合；

文中脚标定义如下：

1.  $t$  代表 LSTM 模型中的时序， $t$  为当前时序，而  $t-1$  或  $t+1$  为前一或后一时序；

2.  $f$  代表任一函数（使用  $f$  指代）代入当前操作或模型；

文中小写字母对象定义如下：

1.  $s$  指代集合  $S$  中的元素，其中  $S$  指代任一集合，如物品集合  $O$  等，他们通常和集合一起出现；
2.  $c$  代表 LSTM 模型中的细胞 (cell)，记录细胞当前状态；
3.  $i$  代表 LSTM 模型中的输入门 (input)，执行 LSTM 模型中的输入操作；
4.  $o$  代表 LSTM 模型中的输出门 (output)，执行 LSTM 模型中的输入操作；
5.  $f$  代表 LSTM 模型中的遗忘门 (forget)，选择 LSTM 细胞的部分状态丢失；
- 6.

文中主要函数定义如下：

1.  $\phi$  代表物品 (object) 的集合；
- 2.

### 3.4 软件界面设计

对于预期的软件效果，我制作了原型图（图 3.1）进行示意。图中，应当出现英文的地方用英文示意，应当出现路径的地方用路径示意，而应当出现中文的地方和说明文字使用中文表述。

其中，界面上方由一组两个单选选项的选择栏和两个数据输入栏组成。选项栏的作用是选择当前需要实现的功能，由“图片翻译语义”和“文本翻译图片”两个选项组成；数据输入栏分别是一个文本输入栏和一个文件选择栏，分别对应着文本和图片的输入选择。

其下是一个明显的按钮，突出的设计可以让用户能轻易明白操作方法，也让用户更有仪式感，能感受到这一操作的划时代意义。

最下方的大区域是输出区域，根据功能的不同可以输出不同的内容。在选择图像翻译语义功能时，会输出一句话，这一行文字和图片将一起显示，方便对比观察；在选择语义生成图像选项时，将输出生成的图片和原语句，也是为了方便用户观察与分享，表意清晰。



图 3.1 软件界面原型图

## 3.5 软件功能设计

功能部分分为图片标注功能以及文字生成图像功能，两个功能我使用了两个结构相对复杂的深度学习模型组合构架的算法，将在下面具体介绍。

### 3.5.1 图片标注方法

#### 3.5.1.1 算法整体模型

在这一模型中，要使用 LSTM 模型进行训练，训练出的模型放在主函数的调用函数中，实现图片翻译为自然语言的功能。

这一模型主要流程由四步组成。

1. 在模型中输入图片，作为输入信息；
2. 由卷积神经网络提取图片信息，形成图片特征信息（即后文编码步骤）；
3. 由注意力机制（attention）对所提取的图片特征信息进行处理，加强或抑制部分区域，作为后续输入 LSTM 的输入信息——在不同时刻，注意力信号会受到

上一次 LSTM 的输出信息的影响，即注意力信号作为 LSTM 神经元细胞的状态，受到输出词语的影响而改变（这也是后文的解码部分）；

4. LSTM 最终输出文本，形成最后的结果。

### 3.5.1.2 编码部分

第一步，要对训练集中的标注编码，形成特征向量。词典中已经预先确定了  $K$  个词语，对于每一行标注  $y_i$ ，可以将其通过词典序号，将句子映射成输入向量，每一个元素的位置意义是序号，即图片相关的类别，数字则是关联度。编码之后生成向量  $\mathbf{y}_i$ ，一起构成输入矩阵。

$$\mathbf{y} = \mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_C, \mathbf{y}_i \in \mathbb{R}^K$$

第二步，对图片编码。使用一个卷积神经网络 (Convolutional Neural Network, CNN) 对图片的特征进行提取，从而形成图片编码。编码好的图片，后续会作为注释向量  $\mathbf{a}$  使用。

$$\mathbf{a} = \{\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_L\}, \mathbf{a}_i \in \mathbb{R}^D$$

### 3.5.1.3 解码部分

解码部分实用的技术是 LSTM，即长短期记忆模型。解码后生成的是标注文本，在预测最后一个词的时候，需要背景向量、前一时刻的隐藏层向量、前一时刻的词。

背景向量  $\hat{\mathbf{z}}_t$  由注意力机制函数和图片注释向量计算得出，并且与时序  $t$  有关，随时序推进而变化。相当于是有选择地传输图片注释向量中的信息，是图片信息的动态表达。

确定一个注意力机制函数  $\phi$  来计算  $t$  时刻的背景向量  $\hat{\mathbf{z}}_t$ 。对于输入的图片注释信息，为了推测这一位置是否是正确的注意力集中点，在式 (3.2) 中定义一个可以归一化的参数  $\alpha_{t,i}$  来表示在  $t$  时刻，位置  $i$  是正确关注点的置信度。这个置信度由，计算背景向量

$$e_{t,i} = f_{att}(\mathbf{a}_i, \mathbf{h}_t) \quad (3.1)$$

$$\alpha_{t,i} = \frac{\exp e_{t,i}}{\sum_{k=1}^L \exp e_{t,k}} \quad (3.2)$$

计算出权重之后，即可使用注意力机制函数计算出背景向量  $\hat{\mathbf{z}}_t$ ：

$$\hat{\mathbf{z}}_t = \phi(\mathbf{a}_i, \alpha_i) \quad (3.3)$$



这里的函数  $\phi$  是利用了软注意力机制<sup>[24]</sup>，这一机制

#### 3.5.1.4 代码实现模块

### 3.5.2 自然语言生成图片方法

这一模型主要是用通过场景生成图像的 GAN 模型来训练，得到的模型放在主函数的调用函数中，实现自然语言生成图片的功能。

#### 3.5.2.1 代码实现模块

**Result:** Write here the result

initialization;

**while** *While condition* **do**

    instructions;

**if** *condition* **then**

        instructions1;

**else**

        instructions3;

**end**

**end**

算法 1: How to write algorithms

## 3.6 本章小结

本章具体地介绍了这一设计中所包含的设计细节及其理论依据，并确定了软件构架的结构，为完成代码构建和实验设计打下了基础。

本章中，确定了以 python 语言的 tkinter 库来写界面，保证界面简单简介易懂；用 CNN 和 LSTM 技术，基于注意力的方法，通过编码-解码的流程来完成图片标注功能；用基于位置场景的方法，由 GNN 和两个判别器构成的 GAN 模型来分步生成图片，最后生成合成图片的方法来实现由文本生成图片的功能。

在下一章里，我会继续记述实验的过程及其表现。

## 4 实验与分析

### 4.1 实验环境

#### 4.1.1 模型训练环境

服务器环境，操作系统为 Ubuntu 16.04.6 LTS (GNU/Linux 4.4.0-142-generic x86\_64)，使用命令行 ssh 工具登入。

内存 64GB，搭配显卡 2080 Ti 四块，由于资源分配原因本设计中实验使用两块。

#### 4.1.2 软件样本运行环境

Macbook Pro 电脑 15.6 寸屏幕版，mid-15 release, A1398 型号；OS X 10.12(Serena High) 系统；集成显卡，内存 16G，没有特别要求。

### 4.2 图片标注实验

#### 4.2.1 模型代码

#### 4.2.2 运行过程

#### 4.2.3 遇到的主要问题与解决方法

### 4.3 自然语言生成图片实验

### 4.4 实验分析与总结

### 4.5 本章小结

## 5 设计意义阐述与展望

### 5.1 我的设计意义

#### 5.1.1 制作了一个简洁易用系统

#### 5.1.2 活用知识、实践了深度学习技术

### 5.2 未来发展方向

## 参考文献

- [1] Cai W, Qiu G. Visual question answering algorithm based on image caption[A]. 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)[C], 2019 : 2076–2079.
- [2] HINTON G E, SALAKHUTDINOV R R. Reducing the dimensionality of data with neural networks[J]. science, 2006, 313(5786) : 504–507.
- [3] GOODFELLOW I, POUGET-ABADIE J, MIRZA M, et al. Generative adversarial nets[A]. Advances in neural information processing systems[C], 2014 : 2672–2680.
- [4] KINGMA D P, WELLING M. Auto-encoding variational bayes[J]. arXiv preprint arXiv:1312.6114, 2013.
- [5] REZENDE D J, MOHAMED S, WIERSTRA D. Stochastic backpropagation and approximate inference in deep generative models[J]. arXiv preprint arXiv:1401.4082, 2014.
- [6] HINTON G E, SEJNOWSKI T J, ACKLEY D H. Boltzmann machines: Constraint satisfaction networks that learn[M]. [S.l.] : Carnegie-Mellon University, Department of Computer Science Pittsburgh, 1984.
- [7] ACKLEY D H, HINTON G E, SEJNOWSKI T J. A learning algorithm for Boltzmann machines[J]. Cognitive science, 1985, 9(1) : 147–169.
- [8] PRADHAN S, HACIOGLU K, KRUGLER V, et al. Support vector learning for semantic argument classification[J]. Machine Learning, 2005, 60(1-3) : 11–39.
- [9] SURDEANU M, HARABAGIU S, WILLIAMS J, et al. Using predicate-argument structures for information extraction[A]. Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics[C], 2003 : 8–15.
- [10] GILDEA D, PALMER M. The necessity of parsing for predicate argument recognition[A]. Proceedings of the 40th annual meeting on association for computational linguistics[C], 2002 : 239–246.

- [11] YAO Y, HUANG Z. Bi-directional LSTM recurrent neural network for Chinese word segmentation[A]. International Conference on Neural Information Processing[C], 2016 : 345 – 353.
- [12] ZHOU J, XU W. End-to-end learning of semantic role labeling using recurrent neural networks[A]. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)[C], 2015 : 1127 – 1137.
- [13] BIRD S, LOPER E, KLEIN E. Natural Language Processing with Python[M]. [S.l.] : O’Reilly Media Inc, 2009.
- [14] NOWOZIN S, CSEKE B, TOMIOKA R. f-gan: Training generative neural samplers using variational divergence minimization[A]. Advances in neural information processing systems[C], 2016 : 271 – 279.
- [15] MIRZA M, OSINDERO S. Conditional generative adversarial nets[J]. arXiv preprint arXiv:1411.1784, 2014.
- [16] ZHANG H, XU T, LI H, et al. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks[A]. Proceedings of the IEEE international conference on computer vision[C], 2017 : 5907 – 5915.
- [17] DONAHUE J, KRÄHENBÜHL P, DARRELL T. Adversarial feature learning[J]. arXiv preprint arXiv:1605.09782, 2016.
- [18] LI Y, SWERSKY K, ZEMEL R. Generative moment matching networks[A]. International Conference on Machine Learning[C], 2015 : 1718 – 1727.
- [19] PERARNAU G, VAN DE WEIJER J, RADUCANU B, et al. Invertible conditional gans for image editing[J]. arXiv preprint arXiv:1611.06355, 2016.
- [20] JIN Y, ZHANG J, LI M, et al. Create Anime Characters with A.I. ![J]. コミックマーケット 92, 2017, ウ 0 5 a.
- [21] ZHANG H, XU T, LI H, et al. Stackgan++: Realistic image synthesis with stacked generative adversarial networks[J]. IEEE transactions on pattern analysis and machine intelligence, 2018, 41(8) : 1947 – 1962.

- [22] JOHNSON J, GUPTA A, FEI-FEI L. Image Generation from Scene Graphs[J/OL]. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018. <http://dx.doi.org/10.1109/CVPR.2018.00133>.
- [23] XU K, BA J, KIROS R, et al. Show, attend and tell: Neural image caption generation with visual attention[A]. International conference on machine learning[C], 2015 : 2048–2057.
- [24] BAHDANAU D, CHO K, BENGIO Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

## 致谢

以简短的文字表达作者对完成论文和学业提供帮助的老师、同学、领导、同事及亲属的感激之情。

## 附录 A 部分代码

A.1 界面代码

A.2 图片标注主函数代码

A.3 图像生成主函数代码