硕士学位论文

基于深度学习的图像标题生成算法及应用

THE RESEARCH AND APPLICATION OF IMAGE CAPTIONING BASED ON DEEP LEARNING

朱丹翔

哈尔滨工业大学

2016 年 12 月

国内图书分类号：TP391.4　　学校代码：10213
国际图书分类号：004.8　　密级：公开

工学硕士学位论文

基于深度学习的图像标题生成算法及应用

硕 士 研 究 生：朱丹翔
导　　师：叶允明教授
申 请 学 位：工学硕士
学　　科：计算机科学与技术
所 在 单 位：深圳研究生院
答 辩 日 期：2016 年 12 月
授予学位单位：哈尔滨工业大学

Classified Index: TP391.4

U.D.C: 004.8

Dissertation for the Master Degree in Engineering

THE RESEARCH AND APPLICATION OF IMAGE CAPTIONING BASED ON DEEP LEARNING

Candidate: Danxiang Zhu

Supervisor: Prof. Yunming Ye

Academic Degree Applied for: Master of Engineering

Speciality: Computer Science and Technology

Affiliation: Shenzhen Graduate School

Date of Defence: Dec, 2016

Degree-Conferring-Institution: Harbin Institute of Technology

摘 要

人工智能是长久以来人类不断探索的一个重要方向，如何让计算机学会人类的能力有着非常重要的意义。得益于计算机并行计算能力的提高和数据爆炸式的增长，产生了大量神经网络类的算法，这类神经网络算法通常网络层数更多，因此又叫做深度神经网络或者深度学习算法。深度学习算法对于复杂的人工智能任务有着惊人的有效性，在多个领域都有所应用。本文的主要研究内容是图像标题生成算法和应用，由于该任务是计算机视觉与自然语言处理两个领域交叉产生的，因此更加复杂，本文将使用深度学习算法对图像标题生成任务中的不同部分进行设计和建模，并且将图像标题生成算法的思想应用在验证码图像识别任务中。

对于图像标题生成任务，本文提出了 past-feeding 和 past-attention 两种算法，分别对不同的网络结构进行改进。第一种 past-feeding 算法，通过加入已输出的词向量的信息作为辅助，联合预测当前时刻的输出词。第二种 past-attention 算法，将多个时刻的注意力向量信息通过 LSTM 建立联系，让注意力向量的生成更加合理，并且将整个模型分为语言信息和图像信息两个部分，使模型更加清晰。本文不仅对模型的大体框架进行详细的阐述，还对相关的公式进行了推导，最后将模型生成标题句子的过程进行可视化，从可视化的图片中可以清晰的看到，算法是如何提取图像特征的，预测标题句子中每一个单词时，注意力是如何变化的。最终实验表明，两种算法在多个指标下均有不同程度的提升。

对于验证码图像标题生成任务，本文提出了 OCR-IC 算法，从图像标题生成算法的角度解决验证码识别问题，并且根据验证码图像的特点对网络结构进行调整。OCR-IC 算法相比于传统算法有着诸多的优势，例如不需要图像分割等人工操作、支持验证码字符变长和模型准确率高等。最终实验表明，OCR-IC 在验证码字符定长和变长两种场景下均有不错的准确率。

关键词：图像标题生成算法；深度学习；验证码识别

Abstract

Artificial intelligence is an important direction that human beings have been exploring for a long time. How to make computers learn human's ability has a vital significance. Benefited from the improvement of computer parallel computing power and data explosive growth, many neural network algorithms came out, this kind of neural network algorithm usually has more network layer, so called deep neural network or deep learning algorithm. Deep learning algorithm have a surprising effectiveness for complex artificial intelligence tasks, and have been applied in many fields. The main research content of this paper is image caption algorithm and its application. The task is more complex because it is the overlapping of computer vision and natural language processing. In this paper, we will design and model the different parts of image caption task using deep learning algorithm, and apply the algorithm in the OCR task.

For the image caption task, two algorithms of past-feeding and past-attention are proposed in this paper, each of them improved different neural network structure. The first past-feeding algorithm, by adding the information of the predicted word embedding as auxiliary, predict current output word. The second past-attention algorithm, by building the relationship of multiple moment attention vector, let the generation of attention vector is more reasonable. And the whole model is divided into two parts of the language information and image information, so that the model is more clear. This paper not only describes the general framework of the model, but also the detail of formulas derivation, finally visualize the process of image caption. From the visualization, we can clearly see how the algorithm extract the image feature and how the attention move when predicting every word in the caption sentence. The final experiment shows that the two proposed algorithms have different degrees of improvement under the different evaluating indicators.

For the verification code image caption task, we propose OCR-IC algorithm to solve this problem in the perspective of image caption. And according to the characteristic of the verification code image, we finetune the network structure to adjust the problem. Compared with the traditional algorithm, the OCR-IC algorithm has many advantages, such as no manual operation of image segmentation, supporting variable length of verification code, high accuracy and so on. The final experiment shows that the OCR-IC algorithm have good accuracy in fixed length and variable length verification code.

Keywords: image caption, deep learning, OCR

目 录

第 1 章 绪　论

1.1 研究背景和意义

随着电脑计算能力的提高以及数据爆炸式的增长，人工智能进入了高速发展的时期，很多原本不可能实现的算法或应用，借助计算机的图形处理器（GPU）强大的并行计算能力和互联网海量的数据得以实现。深度学习作为人工智能的一把利剑，近几年来在语音识别、图像识别、机器翻译等有了突破性进展，本文将基于深度学习方法，对图像标题生成任务进行研究与应用。

图像标题生成任务的目的是输入一张图片后能够自动生成这张图片的标题，输出的标题就是一段描述这张图片的文本，更加通俗的来说，就是给定计算机一张图片，让计算机用自然语言说出它在图片中看到了什么。任务的目的决定了完成这项工作不会简单，在语音识别、图像分类、图像目标检测等人工智能任务中，其问题的本质是对事物的认知，而在图像标题生成任务中，不仅涉及到对事物的认知，还涉及到理解事物之间的联系。例如，在图像分类任务中只需识别图片中的一个目标并给出正确的类标，而在图像标题生成任务中，不仅需要识别图片中的多个目标，还需要理解目标之间的联系，最终组织成一句合理的话来描述图片的内容。此外，由于图像标题生成任务涉及了图像和自然语言两个领域，所以需要同时运用图像和自然语言处理两个方面的知识，并且有效的结合起来用于解决问题。相比之下，本文研究的课题从各个方面都更加有难度，非常富有挑战性。本课题的应用领域也是非常广泛，从技术角度，可以应用在图文搜索、文图搜索、图文翻译中，已有的图像和文字之间的搜索，大多都是基于图片标签的搜索，本质上还是一个自然语言处理和信息检索的问题，本课题是真正意义上建立图像和自然语言之间的联系，让图像和文字之间的检索是基于内容的，而不是基于标签的。从场景角度，可以应用在儿童早教中，教导儿童看图说话，而本文尝试将该技术应用在验证码识别任务中，以更加自然的方式教会计算机看懂验证码图片的内容，并输出验证码对应的正确答案。

深度学习近几年非常火热，得益于其惊人的有效性，深度学习的本质是深层的神经网络，之前研究一直停滞不前很大原因是因为深层神经网络在训练过程中梯度的弥撒和爆炸而导致无法有效的训练，直到 2006 年 Hinton 提出了逐层预训练的方法才一定程度上解决了深层训练问题[1]，随着大量学者重新将目光聚焦到深度学习上，2015 年何凯明博士提出深度残差网络[2]，首次将神经网

络叠到 152 层，最终效果有巨大提升。深度学习的算法包含许多，每种算法都有不同的适用场景，深度信念网络（DBN）适合无监督自学习，卷积神经网络（CNN）适合发现数据的局部相关性，在图像中应用广泛，循环神经网络（RNN）适合时序模型，在语言模型、时间序列等任务中应用广泛，本文将使用多种深度学习算法联合解决图像标题生成问题，不同的算法解决问题中的不同部分。图像标题生成任务的关键点在于如何提取图像信息和生成文本，以及二者之间的有机的融合。图像特征有很多种类，例如颜色分布、纹理特征、形状特征等，比较有名提取方法有 HOG[3]、LBP[4]、Haar[5]等，这些都属于人工设计的特征，而近些年在图像领域大放光彩的卷积神经网络在提取图像特征上表现出强大的力量，可以让计算机自动学习图像特征，不需要人工设计，本文也将使用卷积神经网络作为图像特征提取器。文本生成过程本身是一个时序过程，和分类任务中每个样本预测一个类标不一样，时序模型每次预测会生成多个类标，并且前后有序，有许多传统的算法例如 Ngram、隐马尔科夫模型等可以解决上述问题，不过都有一些自身的局限性，本文将使用深度学习中的循环神经网络来构建模型。本文的研究对于推动人工智能有着重大的意义，如何让计算机像人一样更好的理解图片中的内容，像人一样组织语言将图片内容表述出来，还需要不断的深入的研究，此外研究成果应用场景也很广泛，例如图文检索、图文翻译等，因此本文的研究具有非常重要学术价值与意义。

1.2 国内外相关研究和综述

1.2.1 传统的图像标题生成算法

图像标题生成算法的研究早在十多年前就已经开始出现了，最初的研究还无法直接输出一个句子，只能预测图像中出现了哪些实体。第一个图像注解系统是由 1999 年 Mori 提出的[6]，其算法将图像划分成多个子图，每个子图都能预测关联上一些词，这些词代表了图中出现的实体，最终合并成整个图像关联的词语。之后 Duygulu 提出使用机器翻译的方式[7]来预测图像中包含了哪些实体，先将图像分割成很多个小块，提取小块图像的特征后，使用 Kmeans 聚类，再将聚类后的小块合成大块图像，最终预测每个合并后的大块图像与每个词之间的对应概率后，可以得到整张图像中包含的词语。虽然随后几年出现了很多不错的算法用于预测图像中包含的实体，但是仍然无法很好的预测这些实体在句子中的顺序以及他们之间的关系。

通过图片预测标题句子要比预测一些词要难得多，因为句子不光包含了实体词，还包含动作词、实体的属性词以及实体之间的关系词等，Gupta 和 Davis

表明通过建模实体在图像中的空间关系可以有效的提升预测实体词和词间顺序的准确率[8]。传统的图像标题生成算法一种可行方法是基于模板的，也就是预先定义好一套生成图像标题的模板，例如：A ___ is ___ the ___. 对于句子模板填充问题，李飞飞教授提出只要解决 3 个关键点：what、where、who[9]，具体就是通过识别图片中包含的事件、场景、实体三种词，从而达到填充句子模板的目的，并且表明识别图中的事件词可以通过场景词的推理来有效的提升效果，例如，通过图片中的“雪山”场景可以很容易的预测出“滑雪”事件。类似的，Yao 和李飞飞教授在另一篇论文中发现图片中的人物和人物的动作也是成对出现的[10]，能够得到其中一个就能很好的推理出另外一个。Farhadi 提出了一种打分的过程来衡量图片和句子的相似度[11]，算法的核心思想是构建一个图片和文本的中间表达，在其论文中称之为语义空间，具体来说就是构建三元组<object，action，scene>，将三元组中的目标、动作、场景的词语分别填写到句子中的三个位置，就能得到一句话。为了得到语义空间需要同时学习两个映射：图像到语义空间的映射和句子到语义空间的映射，学习方式主要是通过马尔科夫随机场学习得到。模型训练完毕后，图像和句子同时都可以映射到语义空间，可以很轻易的算出图像和句子的相似度，这种方式的优势是可以很自然的应用到图像-标题检索或者标题-图像检索。Kulkarni 提出了一种包含 2 个阶段的算法[12]，第一个阶段称为内容计划阶段，具体是使用条件随机场算法寻找图像中包含的内容，其中使用了实体、属性和介词的一阶势函数和文本语料的高阶势函数来构建条件随机场，第二个阶段称为表面合理化阶段，将第一阶段的输出结果编码后再解码成自然语言描述，使用 Ngram 算法来解码，模板作为约束。虽然基于模板的方式是一种可行的方法，但是缺点也很明显，不仅局限性大，样式少，不灵活，并且非常依赖人工设计，为了能得到更加丰富的描述，Mitchell 使用 70000 张 Flickr 图片描述构建语法树[13]，最终效果超过基于模板的方法。

1.2.2 基于深度学习的图像标题生成算法

近几年深度学习技术有非常大的进步，在图像标题生成算法中也获得应用并取得了突破性的结果，例如多模算法。多模的意思是模型的输入包含多种形式的数据，例如图像、文本、语音等，算法的主要思想是以多种不同的输入源作为输入，学习一个联合表达向量，用于后续的分类或者回归等。不同的输入源会包含不同的信息，同时也会包含不同的表达方式和相关结构，例如在词袋模型中，文本通常都会被表示为离散稀疏的词频向量，而图像则通常会被表示为像素矩阵，或者是经过特征提取后的实值稠密向量，这种多模态的表达方式会使发现它们之间的高维非线性关系变得更加困难。一个好的多模联合表达通

常要求一个表达向量能够对应一个“概念”，并且在已知一种模态的输入时，能够补全其他缺失模态的输入。

2014 年 Srivastava 和 Salakhutdinov 提出多模深度玻尔兹曼机来学习图像和文本的联合表达[14]，并且指出在有一个或者多个模态输入缺失的情况，模型仍然可以工作，甚至可以使用补全这些缺失模态的输入。同年 Kiros 提出 MLBL 模型[15]，相比于 Srivastava 和 Salakhutdinov 的深度玻尔兹曼机算法，Kiros 使用卷积神经网络提取图像特征，以滑动窗口的形式拼接词向量作为文本上下文特，然后图像特征和文本特征加权求和后生成多模联合特征，最终使用对数双线性模型预测句子的下一个单词，整体算法框架是一个前向神经网络。得益于近几年循环神经网络的发展，有越来越多研究人员开始使用循环神经网络作为文本特征的表示方法， Mao 提出的 m-RNN（多模循环神经网络）模型[16]，基本思想和 Kiros 的模型一致，图像特征同样都是使用卷积神经网络提取，不过对于文本特征的提取，Mao 使用了循环神经网络，最终模型效果在图像标题生成、图像-标题检索、标题-图像检索三个任务中都取得了巨大的提升。

多模算法虽然将图像和文本信息联合表示成一个向量，有利于很自然的应用到两种检索任务当中，但是由于输入包含了图像和文本两种输入源，所以在图像标题生成任务中只能生成当前已有标题句子。在文献[17]中，Vinyals 提出 NIC 模型[17]，可以很自然的解决图像标题生成问题，并且相比于之前的多模算法，NIC 模型对于一张新的图片，可以生成一个新的标题句子来描述该图片。NIC 算法的基本思想和机器翻译的算法思想非常相似，二者都服从于编码器-解码器框架。在机器翻译中，算法的编码器部分是将源语言压缩成一个语义向量，解码器部分将之前的语义向量再解码成一句话，而具体编码器和解码器使用的算法可以根据需要改变。在 NIC 算法中，算法的编码器部分是将图像信息压缩成一个图义向量，使用 VGG 网络提取图像特征作为图义向量，解码器部分将图义向量解码成一句话，使用 RNN 作为解码器，可以看到，整体过程和机器翻译非常类似，所以从这个角度来说，图像标题生成任务也可以被称为图文翻译。2015 年，以 NIC 为基础，同样借助机器翻译的灵感，Xu 提出在图像标题生成任务中加入注意力模型[18]，大大提升了最终效果。注意力模型的基本思想是在预测输出标题句子中的每一个词语时，只关注图像中的某一部分信息，而不是每次都关注整张图片所有的信息，这样预测每个单词的信息来源更加有目标性，从而减少错误的可能性。

尽管深度学习已经在图像分类、目标检测和语音识别等任务中取得了卓越的成就，甚至某些任务中深度学习的能力已经超过了人类的能力，但是在自然语言处理领域中，深度学习还有很长的路要走，归根结底是因为自然语言处理

更多是理解、思考层面，不仅仅只是认知层面，因此还需要研究人员不断努力。图像标题生成任务同时涉及图像和自然语言处理两个领域，因此难度更加大，如何能够很好的识别图像中的实体，以及实体之间的关系，并且以人类可读的自然语言的方式输出出来，需要更加深层的神经网络和更加优良的网络架构来发现图像和文本数据中复杂的模式和关系，并且还需要更加庞大的数据集和计算机更加强大的并行计算能力作为支撑。

1.3 问题的总结与分析

综上所述，深度学习是发现复杂的数据模式的重要技术，在图像、文本、语音领域都发挥着巨大的作用，而传统的图像标题生成算法是基于模板的，太过依赖于人工设计，并且局限性大，效果也不尽如人意，如何将深度学习合理的融入图像标题生成任务中，发挥深度学习端到端学习的优势，提升最终生成的标题句子质量，增强标题句子与图像主题的匹配性和可读性，对推动图像和自然语言处理领域，乃至整个人工智能领域都有重要的意义。因此，本文对图像标题生成算法、深度学习、图像和自然语言处理的国内外研究进行了调研，根据调研结果，本文将图像标题生成算法的研究和在验证码识别上的应用分为两个子课题。

在图像标题生成算法的研究中，目前多数研究人员都着眼于标题句子中的单词和图像中的内容的对应上，尚缺乏对已输出的图像内容对于后续输出的影响的研究。如果在模型解码阶段预测每一个词的时，模型无法得知已经输出过图像中的哪些内容，则有可能导致图像中的某一部分内容被重复输出，最终使输出结果质量下降。

在验证码识别的应用上，本文将使用图像标题生成算法的思想，实现验证码识别系统。传统的验证码识别系统大多数都是需要先使用数字图像处理的方法先对验证码图片进行处理，这样不仅需要人工操作，系统的可移植性也不高。本文基于图像标题生成算法，将验证码图片作为输入，验证码的字符作为标题句子，将验证码识别问题转化为验证码标题生成问题，改进传统验证码识别需要人工操作的缺点，充分发挥深度学习端到端学习的优势。

对于本文的研究内容，是深度学习、图像和自然语言处理三个领域的知识交叉的结果，目前国内外的研究才刚刚起步，基于深度学习的图像标题生成系统还没有一个通用完整的解决方案，是一个仍需要研究人员投入大量工作的课题，不过得益于大数据时代产生的大量数据和逐渐强大的 GPU 并行计算能力，为该课题的研究奠定了坚实的研究基础。

1.4 本文主要工作

综上所述，传统图像标题生成算法有着不灵活、依赖人工设计等诸多缺点，在实际工作中已经逐渐被摒弃，虽然深度学习技术在多个领域都表现出巨大的潜力，但是由于本身图像标题生成任务的复杂性，导致目前以深度学习为基础对图像标题生成任务的研究还处于起步状态。本文将基于深度学习的多种算法，对图图像标题生成问题进行探索，并且最终将图像标题生成算法应用在验证码识别问题中。具体来说，本文的主要工作包含以下两个方面：

（1）对于图像标题生成算法的研究，本文将在 NIC 算法和注意力模型的基础上进行。模型以编码器-解码器作为整体框架，编码器使用 VGG 卷积神经网络将图像信息编码为图义向量，解码器使用 LSTM 将图义向量循环解码为一个句子，LSTM 是循环神经网络中的一种，相比于原始的 RNN，LSTM 能够记住的前文信息更加多，并且还能很好的解决梯度弥散和爆炸问题。本文尝试将注意力模型在不同时刻生成的注意力向量建立联系，并且将模型中图像部分和文本部分分离后再融合，最终提升输出的标题句子的质量。

（2）在验证码识别的应用上，本文以图像标题生成角度，将验证码图片作为输入，验证码数字作为标题句子，建立验证码标题生成系统。由于验证码图片和自然场景图片有较大差异，所以在编码器的阶段不使用 VGG 网络，而是使用更加简单的卷积神经网络，并且由于验证码标题的特殊性，每个字符之间没有前后关联，所以去除了解码器中文本 LSTM 部分，留下了图像注意力 LSTM 部分。相较于传统的验证码识别系统，本文的方法能够省去人工操作的步骤，整体训练过程端到端，并且能够取得不错的效果。

1.5 本文组织结构

本文将以深度学习作为基础，对图像标题生成任务进行模型设计、模型改进和训练、结果可视化以及相应的系统实现，并且以一个全新的角度对验证码识别问题进行分析、模型设计以及系统的实现。本文的整体的章节安排将围绕上述的要点进行展开论述，具体每个章节的内容安排如下：

第 1 章为绪论，叙述了图像标题生成课题的背景以及意义，并且，从技术角度简单介绍了近几年非常火热的深度学习的概念以及如何将深度学习融入到本课题中，从应用角度介绍了该课题可能的应用场景。同时介绍了本课题的国内外研究现状，主要从传统做法和深度学习做法的两个历史发展过程进行阐述。最后介绍了本课题所要研究的问题以及主要工作内容，主要包含了算法的研究以及验证码识别场景的应用。

第 2 章为深度学习相关的基础知识，主要介绍了本文用到的几种神经网络，包括最基本的多层感知机、卷积神经网络和循环神经网络，并且分别介绍了卷积神经网络中的卷积和池化操作以及 VGG 网络，还介绍循环神经网络中原始的 RNN 和性能更优秀的 LSTM。

第 3 章为基于深度学习的图像标题生成算法的研究，主要阐述了算法的整体框架，以及本文提出的基于 past-feeding 和基于 past-attention 的图像标题生成算法，然后简要的介绍了实验环境与数据和评价指标，最后进行实验结果的分析与可视化。

第 4 章为验证码图像标题生成系统的设计与实现，先介绍了数据来源与预处理，然后详细阐述了系统整体框架的设计与实现以及模型的训练方法，专门针对验证码识别问题设计了图像编码器和验证码字符解码器，没有直接使用第 2 章介绍的 VGG 网络作为编码器，最后进行了实验结果分析与可视化。

最后结论部分总结了本课题的研究成果、期间遇到的问题以及存在的不足，为未来的研究者提供经验和参照。

第 2 章 深度学习的相关基础知识

本文的主要围绕基于深度学习的图像标题生成算法的研究，在系统阐述图像标题生成算法前，本章先介绍深度学习相关的知识，主要内容包含了最基础的多层感知机，在图像领域大红大紫的卷积神经网络，以及在自然语言处理领域得到广泛应用的循环神经网络。

2.1 多层感知机

尽管深度学习在多种人工智能任务中都表现卓越，但是归根结底来说，深度学习就是神经网络。一个标准的神经网络包含了许多简单的相互连接的小处理器，这些小处理器通常都被称为神经元，除了输入层的神经元，其他层的每个神经元都会直接与前一层所有神经元相连接，不断的在已有的神经元的基础上叠加下一层神经网络，就会形成多层神经网络，这种多层神经网络也被称为多层感知机。图 2-1 是一个最简单的三层感知机，包含了输入层、一层隐藏层和输出层。

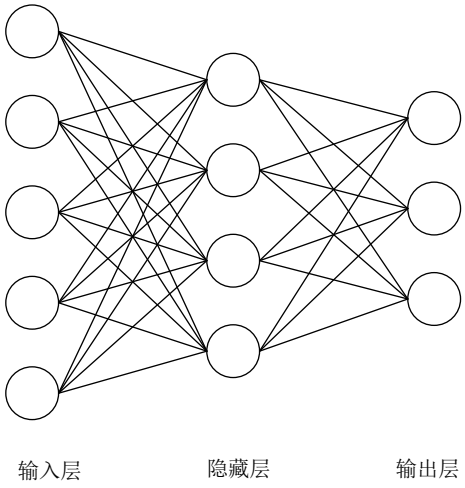


图 2-1 三层感知机

多层感知机中每一层的神经元的激活值都由前一层的神经元的激活值加权求和并加上偏置项，然后输入激活函数后得到，如公式（2-1）所示，其中激活函数的定义可以根据实际问题而更改，典型的激活函数包含 sigmoid、tanh 和 ReLU 等，值得注意的是，本文为了让公式更加简洁易懂，默认所有公式变量都使用矩阵表示，因此权值 $W^{(l-1)}$ 中并没有使用下标来表示某一个权值，而是直

接使用$W^{(l-1)}$表示第$l-1$层连接第l层的所有权值，具体实现时可以用矩阵相乘的方式计算，公式（2-2）表示 ReLU 激活函数。

$$a^{(l)} = \delta(W^{(l-1)} \cdot a^{(l-1)} + b^{(l)}) \tag{2-1}$$

$$\delta(x) = \max(0, x) \tag{2-2}$$

深度学习之所以沉寂多年，很大的原因来自于多层神经网络训练非常困难，因为当神经网络层数很多时，训练过程中的梯度会随着层数的增多而急剧下降或者急剧上升，也就是通常所说的梯度的弥散和爆炸，根本原因就是公式（2-3）和（2-4）导致的。当需要求第一层神经元的导数时，根据链式法则，可以被分解为中间每一层对前一层神经元的导数的乘积，相邻层的导数通过公式（2-4）可以被进一步表示为激活函数的导数与权值的乘积，而激活函数如果是非线性函数，例如 sigmoid 函数，则其导数的最大值为 0.25，所以相邻层的导数值很容易小于 1，当小于 1 时，由于是连乘的关系，随着逐渐向后传播，梯度值会以指数级的程度减小，最终传播到第一层时，已经几乎为零了，这就是所谓的梯度弥撒，反之，大于 1 时则会引起梯度爆炸。

$$\frac{\partial Loss}{\partial a^{(1)}} = \frac{\partial Loss}{\partial a^{(L)}} \cdot \frac{\partial a^{(L)}}{\partial a^{(L-1)}} \cdots \frac{\partial a^{(2)}}{\partial a^{(1)}} \tag{2-3}$$

$$\frac{\partial a^{(L)}}{\partial a^{(L-1)}} = \delta \cdot (1-\delta) \cdot W^{(L-1)} \tag{2-4}$$

2.2 卷积神经网络

深度学习之所以能够在图像领域的各大比赛中独占鳌头，很大的功劳都要归于卷积神经网络惊人的有效性。1998 年 Lecun 提出卷积神经网络并成功应用在手写识别库系统中[19]，算法灵感来源于大脑视觉皮层的工作机制。卷积神经网络不仅有惊人的有效性，同时由于包含了局部感受野和权值共享两大特点，能够大大减少模型中权值的个数。卷积神经网络之所以比较适合图像任务，是因为人类在识别图像时通常都是关注于图像中的轮廓纹理信息，不会注意到图像中像素级别的信息，而卷积神经网络恰恰能够发现像素与周围像素之间的局部相关性，这种局部相关性组合起来就是图像的轮廓纹理等高级抽象信息。此外，在图像领域中好的图像特征需要符合 3 个条件，即图像在经过旋转、平移和缩放后，提取的图像特征仍能很好的表达原图，其中平移不变性可以通过卷积操作来达到，旋转和缩放不变性通过池化操作来达到，不断的对图像施加卷积和池化操作就能够提取到非常好的高维抽象特征，将这些图像特征用于分类或者聚类能够很好的提升最终模型效果。

2.2.1 卷积操作

通常卷积神经网络中包含两种操作，卷积和池化，本节将介绍卷积的概念以及定义。卷积这个名词最早是来源于信号系统理论中，而后被许多数学家发扬光大，直到 1998 年 Lecun 将卷积的思想应用到神经网络中提出了卷积神经网络，为现代机器视觉领域做出了巨大的贡献。

$$h(x)=\int_{-\infty}^{\infty} f(\tau)g(x-\tau)d\tau \tag{2-5}$$

公式（2-5）是卷积在数学中的定义，可以看到，卷积的本质其实就是一个函数在另一个函数上的加权叠加。而对应到神经网络中，卷积操作的第一个函数是图片或者特征图，第二个函数是卷积核，将特征图在卷积核上加权叠加就能生成新的特征图，当卷积核有多个就会对应多张特征图。

$$a_j^{(l)}=\delta\left(\sum_{i\in M_j} a_i^{(l-1)} * K_{ij}^{(l)} + b_j^{(l)}\right) \tag{2-6}$$

公式（2-6）是卷积在神经网络中的前向传播的公式[20]，其中 δ 是激活函数，M_j 表示连接到第 l 层第 j 个特征图的第 $l-1$ 层的特征图集合，$a_i^{(l-1)}$ 表示第 $l-1$ 层第 i 个特征图，$K_{ij}^{(l)}$ 表示第 $l-1$ 层第 i 个特征图与第 l 层第 j 个特征图之间的卷积核，b_j^l 表示与第 l 层第 j 个特征图相关的所有卷积核对应的偏置项。

图 2-2 展示了一个卷积神经网络样例，包含了 2 个卷积层和 2 个池化层，可以看到原始输入包含三个通道，每个通道的图像都是 32*32 大小的，经过一层 5*5 大小卷积核的卷积操作后，生成了 28*28 的特征图。

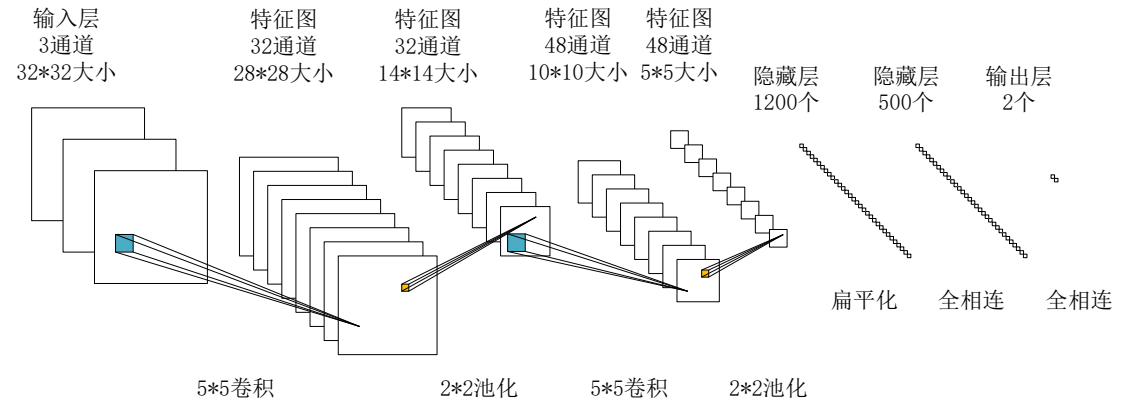


图 2-2 卷积神经网络（2 层卷积+2 层池化）

卷积神经网络中的卷积核可以自动发现图像中的细节纹理特征，这些细节纹理特征不断组合可以生成更加高级的抽象特征，图 2-3 展示了图像在卷积神经网络中不同层的特征，可以从图中发现，底层的图像特征是图像的纹理信息，

随着层数的增多，特征抽象程度逐渐增加，在第三层图像特征中已经能够发现人类和动物的头部轮廓特征，这种高级抽象特征对于最终的图像分类任务很有帮助[21-23]。

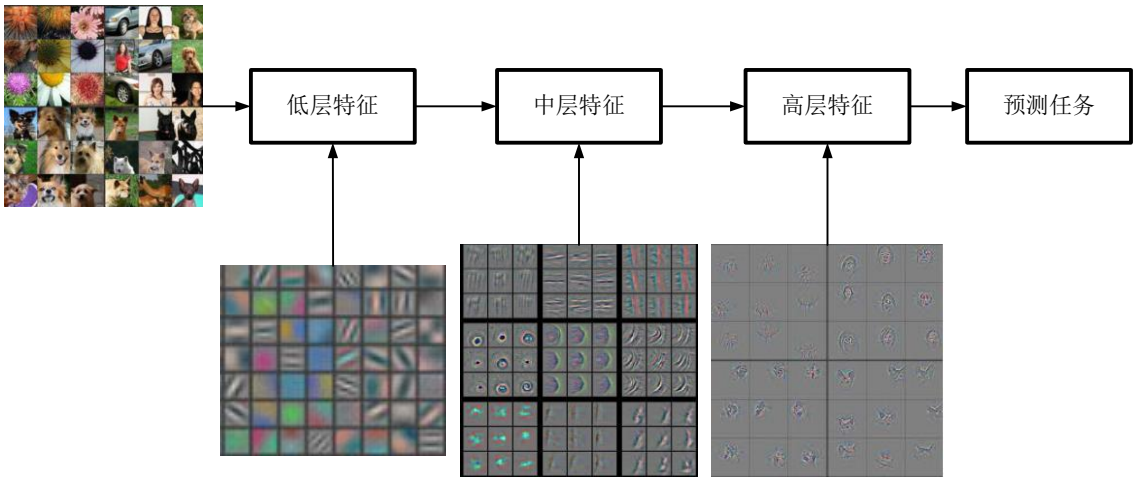


图 2-3 分层图像特征

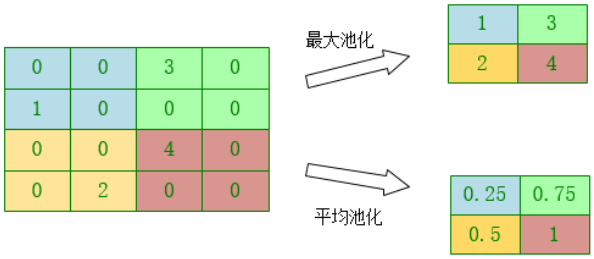


图 2-4 最大池化与平均池化

2.2.2 池化操作

池化是卷积神经网络中的第二种常见的网络层，在一些文献中池化也常常被称为下采样或者子采样操作。池化操作常见的类型也分为两种，分别为最大池化和平均池化，两种池化操作分别有不同的作用。一般来说，平均池化能够更多保留图像背景信息，最大池化能够更多保留图像的纹理信息[24]。

$$a_j^{(l)} = \delta(\beta_j^{(l)} down(a_j^{(l-1)}) + b_j^{(l)}) \tag{2-7}$$

公式（2-7）是池化在卷积神经网络中前向传播公式，其中 β 表示乘积偏置，b 表示加法偏置，$down$ 是池化函数，也就是上文提到的最大池化和平均池化。通过池化操作不仅能够使提取的图像特征具有良好的性质，还能大大减少模型的参数和计算量，在图 2-2 中所示的卷积神经网络包含了 2 次池化操作，每一次池化操作都使特征图的大小缩小了一半，例如在第一次池化过程中，特征图的大小从 28*28 变为了 14*14，直接使后续操作的时间复杂度减少了一半，参

数总数也随之减少。图 2-4 展示了最大池化和平均池化的计算过程的样例，其中不同颜色代表不同的从不同的池化域中计算得到的。

2.2.3 VGG 网络

卷积神经网络是对通过卷积和池化来提取特征的神经网络的统称，但是具体网络结构却是层出不穷，例如用于手写识别的 LeNet，2012 年在 ImageNet 比赛中获得冠军的 AlexNet[25]，2014 年提出的 GoogLeNet[26]和 VGG[27]网络，以及 2015 年提出的 ResNet 等，每个网络都有不同的特点和使用场景。本文将使用 VGG 网络作为图片特征提取器，虽然发表时间距离更近的 ResNet 的效果更加好，但是相应消耗的显存以及计算时间也非常巨大，本课题综合考虑时间和硬件资源问题，采用效果也很杰出的 VGG 网络作为图像特征提取器。

相比于之前的提出的卷积神经网络结构，VGG 网络最大的特点就是“深”，其网络层数最高达到 19 层，最终在 ImageNet 数据集上的分类效果有巨大的提升，top-1 错误率达到 23.7%，top-5 错误率达到 6.7%，而 2012 年的冠军 AlexNet 的 top-1 错误率为 38.1%，top-5 错误率为 16.4%，相比减少了一半左右的错误率。VGG 之所以效果如此惊人，就是依赖于堆叠了足够多层数的神经网络，同年一起提出的 GoogLeNet 也不约而同加深了网络，但是为什么网络层数越多能导致最终效果能越好呢？一个简单的例子就能说明，3 层卷积核大小为 3*3 的卷积层堆叠起来和 1 层卷积核大小为 7*7 的卷积层对输入图像进行卷积，会得到相同大小的特征图，但是不同之处有三点：第一，小卷积核相对于大卷积核能够捕获图像更佳细节的信息；第二，3 层小卷积核的卷积层会对输入数据进行 3 次非线性变换，而 1 层大卷积核只能进行 1 次非线性变换，最终导致前者形成的决策函数更易分类；第三，多层小卷积核的堆叠比一层大卷积核的参数要少很多，假设输入和输出层的都是 N 个通道，3 层 3*3 小卷积核堆叠的参数总量为$3\cdot(3^2\cdot N^2)=27N^2$，同时，1 层 7*7 大卷积核的参数量为$1\cdot(7^2\cdot N^2)=49N^2$，可以看到，前者相对于后者参数总量节省了约 45%。因此，在计算资源允许的情况下，应该尽可能的将大卷积核分解为多个小卷积核使用。

VGG 网络的作者在论文中由浅到深提出了五种网络结构，如表 2-1 中所示，A-E 五种网络包含不同数量的卷积层，其中网络 E 的层数最多同时效果也最佳。虽然实验表明层数越多，最终效果越好，但是层数越多也随之带来其他的问题，例如训练时间过长、梯度弥散或爆炸导致训练不稳定问题，为了解决训练不稳定问题，作者在论文中使用了权值初始化的方法来训练神经网络。本课题综合考虑数据规模、训练时间和硬件资源问题，最终选择了网络 D，一共包含 13 个卷积层，整体分为六个部分，前两个部分各自包含 2 个 3*3 的卷积层，分别输

出 64 和 128 个特征图，接着三部分各自包含 3 个 3*3 的卷积层，分别输出 256、512 和 512 个特征图，最后一部分包含 3 个全相连层和一个 softmax 层。本课题所提取的图像特征是第五部分的第三个卷积层特征，和第六部分第一个全相连层的特征，然后将提取的特征不断解码成一句话当做图像的标题。

表 2-1 VGG 网络的五种网络结构

网络配置				
A	B	C	D	E
11 个权值层	13 个权值层	16 个权值层	16 个权值层	19 个权值层
输入(224 * 224 RGB 图片)				
3*3 卷积-64	3*3 卷积-64 3*3 卷积-64	3*3 卷积-64 3*3 卷积-64	3*3 卷积-64 3*3 卷积-64	3*3 卷积-64 3*3 卷积-64
最大池化				
3*3 卷积-128	3*3 卷积-128 3*3 卷积-128	3*3 卷积-128 3*3 卷积-128	3*3 卷积-128 3*3 卷积-128	3*3 卷积-128 3*3 卷积-128
最大池化				
3*3 卷积-256 3*3 卷积-256	3*3 卷积-256 3*3 卷积-256	3*3 卷积-256 3*3 卷积-256 1*1 卷积-256	3*3 卷积-256 3*3 卷积-256 3*3 卷积-256	3*3 卷积-256 3*3 卷积-256 3*3 卷积-256 3*3 卷积-256
最大池化				
3*3 卷积-512 3*3 卷积-512	3*3 卷积-512 3*3 卷积-512	3*3 卷积-512 3*3 卷积-512 1*1 卷积-512	3*3 卷积-512 3*3 卷积-512 3*3 卷积-512	3*3 卷积-512 3*3 卷积-512 3*3 卷积-512 3*3 卷积-512
最大池化				
3*3 卷积-512 3*3 卷积-512	3*3 卷积-512 3*3 卷积-512	3*3 卷积-512 3*3 卷积-512 1*1 卷积-512	3*3 卷积-512 3*3 卷积-512 3*3 卷积-512	3*3 卷积-512 3*3 卷积-512 3*3 卷积-512 3*3 卷积-512
最大池化				
全连接-4096				
全连接-4096				
全连接-1000				
Softmax 层				

2.3 循环神经网络

循环神经网络是深度学习的第二大利器，在语音和自然语言处理领域中有广泛的应用[28-31]。1990 年 Elman 第一次在论文中提出循环神经网络[32]，用其解决异或序列预测问题，虽然任务很简单，但是已足够证明循环神经网络有能力解决时序数据建模问题。随着问题逐渐变得复杂，研究人员发现普通的循环神经网络并不能记住序列中相隔很远的信息，并且相比于普通的神经网络，循环神经网络更不容易训练，更容易受到梯度弥散和爆炸问题的影响。为了解决普通神经网络“短视”和梯度不易训练的问题，1997 年 Hochreiter 提出了 LSTM 循环神经网络[33]，通过加入上下文信息存储器、输入门、输出门和遗忘门，让网络更加容易训练，并且能够记住更远的信息。原始的循环神经网络和 LSTM 都属于循环神经网络，本文为了避免混淆，将原始的循环神经网络称为 Elman-RNN，当出现循环神经网络或者 RNN 时是泛指包含 Elman-RNN 和 LSTM 等能够建模时序数据的神经网络。

2.3.1 Elman-RNN

Elman-RNN 的结构和标准的神经网络很相似，不同之处在于 Elman-RNN 的隐藏层输入不仅包含当前时刻的数据输入，还依赖前一个时刻隐藏层的输出，使得后时刻的输出可以用到前时刻的信息。如果将整个网络按时间顺序展开，可以发现 Elman-RNN 的网络结构变成一连串多层神经网络前后依次连接，如图 2-5 所示。从图中可以看到，如果去除隐藏层之间的联系，单独就一个时刻而言，其实就是一个三层的神经网络，以 x 作为输入，h 为隐藏层，y 为输出，每个时刻的输出都与上下文无关，但是真实的很多场景中，每个时刻的输出和上下文是高度相关的，例如在文本领域中，某一个词的词性和它前一个词与后一个词是高度相关的，正因为语言有这种相关性，人类说话才会有语法规则。

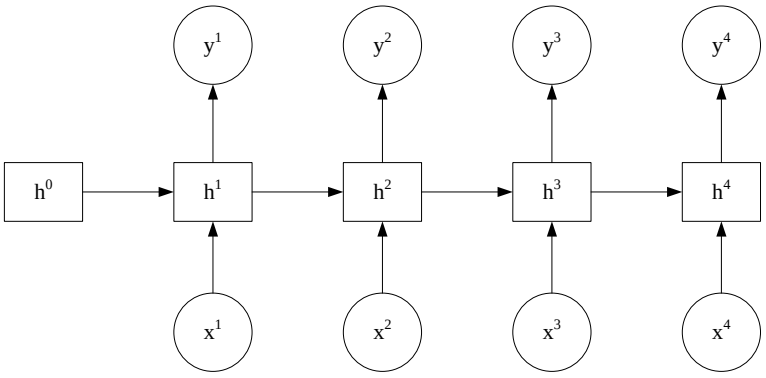


图 2-5 按时间顺序展开的 Elman-RNN

Elman-RNN 不仅能够完成普通多层神经网络无法做到事情，而且其公式推导非常简单，如公式（2-8）和（2-9），其中上标 t 代表第 t 时刻，W_{x2h} 代表从输入层到隐藏层的权值，W_{h2h} 代表从前一个时刻的隐藏层到当前时刻的隐藏层的权值，W_{h2y} 代表从隐藏层到输出层的权值，b_h 和 b_y 分别为隐藏层和输出层的偏置，并且每个时刻都共享这 5 个参数。 从公式中可以发现，每个时刻的输出取决于当前时刻的隐藏层的信息，而当前时刻隐藏层的信息取决于当前时刻的输入和前一个时刻隐藏层的信息，稍微推演可以得知，每个时刻的输出被当前时刻以及之前所有时刻的输入所影响，因此 Elman-RNN 是一种建模时序数据的可行的方法。

$$h^{(t)} = \tanh(W_{x2h} \cdot x^{(t)} + W_{h2h} \cdot h^{(t-1)} + b_h) \tag{2-8}$$

$$y^{(t)} = soft\max(W_{h2h} \cdot h^{(t)} + b_y) \tag{2-9}$$

虽然 Elman-RNN 能够利用前文信息支持当前时刻的决策，但是这种网络结构却非常难以训练，因为如果从另一种角度来看，Elman-RNN 其实是一种非常深的神经网络，越是深的神经网络，越易受到梯度弥散和爆炸问题的影响，导致训练过程非常不稳定，虽然有研究人员提出一定的对策来解决梯度弥撒和爆炸问题，例如梯度截断等，但是并没有从根本上解决梯度爆炸问题，直到 LSTM 出现，使得循环神经网络的训练难度得到一定的缓解。

2.3.2 LSTM

从上一节的最后一部分对 Elman-RNN 的介绍中可以得知，Elman-RNN 存在训练困难的问题，归根结底是因为随着网络的不断传播，梯度急剧减小或者增大，而梯度的剧烈变化是因为非线性激活单元的取值很容易饱和，在取值饱和的地方梯度值会非常小。为了解决循环神经网络中梯度弥散和爆炸问题，Hochreiter 提出 LSTM，在 Elman-RNN 的基础上，加入输入门、输出门和记忆存储器，后续其他的研究人员又加入遗忘门，联合组成目前广泛应用于时序数据建模的 LSTM 算法，相比于 Elman-RNN 网络效果更加优秀。

如图 2-6 所示，LSTM 的网络结构看似比 Elman-RNN 要复杂很多，但是如果将 LSTM 当做一个特殊的神经激活函数（图中的虚线框部分），那么其实 LSTM 和 Elman-RNN 的网络结构很相似。LSTM 的复杂和改进之处在于图中的虚线框部分，其中 i、o、f 分别为三种门电路，用于控制信息的流入流出，每种门电路都由当前的输入和前一时刻的隐藏层所决定，并使用 *sigmoid* 函数将值域压缩至[0,1]，用于表示信息经过门电路时的通过率，具体形式如公式（2-10）至（2-15）所示。图中 g 的作用是将数据进行非线性变换，非线性激活函数使

用 tanh，具体如公式（2-13）所示。图中 c 代表记忆存储器，是 LSTM 的核心部件，其作者在论文中提出，正因为有了记忆存储器，梯度才能以常数级别向后传播，不会受到非线性激活函数的所导致的梯度指数级衰减，从而使 LSTM 能够记住相隔更远的信息，具体的计算公式如（2-14）所示。最终 LSTM 的隐藏层的输出由输出门和当前记忆存储器中的内容所决定，如公式（2-15）。

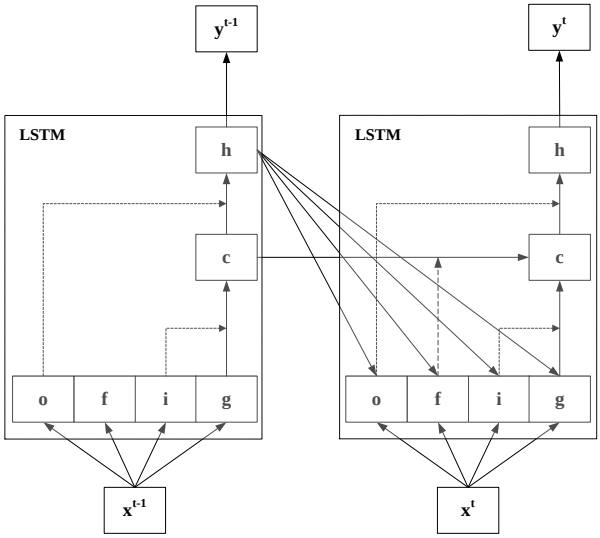


图 2-6 按时间顺序展开的 LSTM

LSTM 的核心计算公式就是（2-10）至（2-15），但是由于公式过于冗长，为了在使用 LSTM 时公式推导更加简洁明了，本文在后续章节中，直接使用公式（2-16）当整个 LSTM 的公式集合，忽略内部细节的公式推导。从公式（2-16）中可以发现，LSTM 在每个时刻的输入只依赖于 $x^{(t)}$、$c^{(t-1)}$、$h^{(t-1)}$，虽然内部结构由门电路组成显得非常复杂，但是宏观来看很简洁，和 Elman-RNN 非常相似。此外，从代码实现的角度来看，也应该将 LSTM 封装成公式（2-16），对外暴露的输入输出接口很简洁，复杂的计算封装在函数体内部。

$$i^{(t)} = sigmoid(W_{x2i} \cdot x^{(t)} + W_{h2i} \cdot h^{(t-1)} + b_i) \tag{2-10}$$

$$o^{(t)} = sigmoid(W_{x2o} \cdot x^{(t)} + W_{h2o} \cdot h^{(t-1)} + b_o) \tag{2-11}$$

$$f^{(t)} = sigmoid(W_{x2f} \cdot x^{(t)} + W_{h2f} \cdot h^{(t-1)} + b_f) \tag{2-12}$$

$$g^{(t)} = \tanh(W_{x2g} \cdot x^{(t)} + W_{h2g} \cdot h^{(t-1)} + b_g) \tag{2-13}$$

$$c^{(t)} = f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot g^{(t)} \tag{2-14}$$

$$h^{(t)} = o^{(t)} \odot \tanh(c^{(t)}) \tag{2-15}$$

$$c^{(t)}, h^{(t)} = LSTM(x^{(t)}, c^{(t-1)}, h^{(t-1)}) \tag{2-16}$$

LSTM 的核心思想与近几年提出的 Highway 网络[34]和 Resnet 有异曲同工之妙，都是使梯度有另一条路径可以向后传播，并且在该路径中，梯度是以常数

级别变化。在 LSTM 的后续研究中[35-40]，有研究人员提出 GRU 算法[41-42]，和 LSTM 的主要思想一致，但是只包含更新门和重置门，参数数量也少一些，在机器翻译任务中有更好的表现。此外，研究人员为了探索什么样的循环神经网络结构能达到更好的效果，使用超参数优化的算法，尝试了 5000 多种 LSTM[43]，这些变种 LSTM 和原始 LSTM 主要区别有以下几种：

1) 是否有输入门
2) 是否有输出门
3) 是否有遗忘门
4) 是否有输入激活函数
5) 是否有输出激活函数
6) 是否有 Peepholes 连接[44]
7) 输入门和遗忘门是否成对出现
8) 是否给所有的门都加上循环连接

通过以上几种条件组合以及超参数的搜索，一共尝试了 5000 多种 LSTM，最终实验结果发现，对 LSTM 效果影响最大的是遗忘门和输出门的激活函数，这说明记忆存储器除了有常数化误差流的作用，对旧信息的遗忘能力和重要信息的存储能力也对 LSTM 整体效果非常重要。此外，如果输入门和遗忘门成对出现时，输出门的重要性就会下降许多，这也解释了为什么 GRU 中删除了输出门后效果并没有下降。

2.3.3 注意力模型

如果关注深度学习最近几年的研究成果，会发现注意力模型在多个领域都被应用并且对最终结果有很大的提升[45-50]。深度学习中的注意力模型和字面上的意思一样，就是代表观察或者理解事物时的注意力，例如，人脑在识别一张图像中猫的特征时，会很自然的将目光聚集在图片中有猫的地方，又例如，在英汉句子翻译任务中，翻译输出中文的每一个单词都会对照英文句子中的某个单词。在本课题的研究中也将加入注意力模型，在解码输出标题句子时，某个单词都会对照图像中的某一部分，例如解码输出单词 dog 时，应该将注意力集中在图像中狗的部分，而忽略图像中其他的部分的信息，使解码过程可解释性更加强。

目前注意力模型在编码器-解码器框架中应用较为广泛，大致分为两种：软注意力模型和硬注意力模型。硬注意力模型主要是使用强化学习的方法进行训练，并且将注意力严格限制在一个很小的范围内，而软注意力模型可以使用梯度下降训练，并且生成一个注意力向量代表注意力的分布，注意力的多少由分

布决定，不会严格控制在一个小范围内，本文也将选择软注意力模型作为图像注意力模块的算法基础。

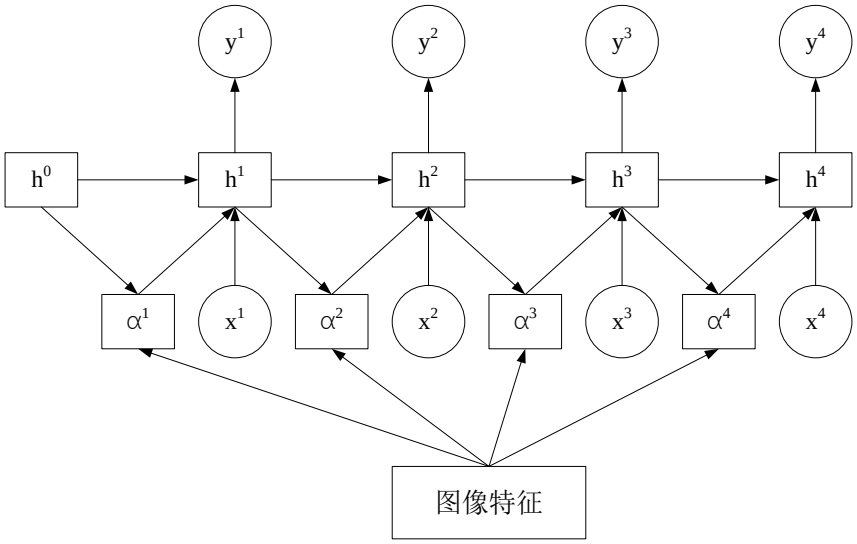


图 2-7 注意力模型

图 2-7 是注意力模型在图像标题生成算法中的应用的网络结构图，其中，α 就是前文所述的注意力向量，隐藏层 h 的计算可以根据自己的需要使用不同的算法，例如 Elman-RNN、LSTM、GRU 等，本文所用的算法为 LSTM。从图中可以总结出以下两点：（1）每个时刻的注意力向量的输入包含 2 个，前一时刻的隐藏层信息和图像特征，具体如公式（2-17）至（2-19）所示，其中 a_i 代表图像特征图第 i 个位置的向量，例如当图像有 512 个 14*14 的特征图时，a_i 则代表特征图中第 i 个点处的 512 维向量，$\alpha_i^{(t)}$ 表示 t 时刻第 i 个位置的注意力值，当特征图的大小为 14*14 时，α 就有 $14*14=196$ 个注意力值；（2）每个隐藏层的输入包含 3 个，前一时刻的隐藏层信息、当前时刻的输入和经过注意力向量作用后的图像特征，具体如公式（2-20）至（2-21）所示，其中 $z^{(t)}$ 表示 t 时刻经过注意力向量作用后的图像特征，其实就是图像特征与对应的注意力向量相乘后求和，为了保证图（2-7）的简洁性，并没有在图中表示出 $z^{(t)}$。此外值得注意的是，公式（2-21）的 LSTM 中有 4 个参数，相比前一节 LSTM 公式推导中多了一个输入，本质上内部公式并没有变，因为可以将 $x^{(t)}$ 和 $z^{(t)}$ 两个矩阵拼接起来当做一个输入，这样就严格符合 LSTM 的参数要求。

$$p_i^{(t)} = \tanh(W_{a2p} \cdot a_i + W_{h2p} \cdot h^{(t-1)} + b_p) \tag{2-17}$$

$$e_i^{(t)} = W_{p2e} \cdot p_i^{(t)} + b_e \tag{2-18}$$

$$\alpha_i^{(t)} = soft\max(e_i^{(t)}) = \frac{\exp(e_i^{(t)})}{\sum_{k=1}^{L} \exp(e_k^{(t)})} \tag{2-19}$$

$$z^{(t)} = \sum_{i=1}^{L} \alpha_i^{(t)} \cdot a_i \tag{2-20}$$

$$c^{(t)}, h^{(t)} = LSTM(x^{(t)}, z^{(t)}, c^{(t-1)}, h^{(t-1)}) \tag{2-21}$$

同样，为了保证后面章节公式推导简单明了，与前一节叙述 LSTM 时对公式的封装一样，本文使用公式（2-22）作为软注意力算法中注意力向量计算的封装，公式中省略了下标 i，表示通过公式（2-22）一次性计算出特征图中每个点的注意力值。

$$\alpha^{(t)} = ATT(a, h^{(t-1)}) \tag{2-22}$$

2.4 本章小结

本章主要介绍了本课题会使用到的多种深度学习算法，具体包括多层感知机、卷积神经网络和循环神经网络，叙述了每个算法的概念定义和公式推导，并且在多个章节中都着重讲述了梯度的弥散和爆炸问题。后续章节会在本章知识的基础上，进行算法的设计以及应用。

第 3 章 基于深度学习的图像标题生成算法

本章的主要内容是基于深度学习的图像标题生成算法的研究，在已有的算法上提出改进。算法使用编码器-解码器框架作为基础，并提出 past-feeding 算法改进标题的全局解码质量，同时，针对使用注意力模型的解码器，本文提出一种新的 past-attention 模型，相比于原始注意力模型，新的模型在生成每一个时刻的注意力向量时加入了过去注意力向量的信息，并且将整体模型分为语言部分和图像部分，最后生成标题句子时，将语言信息和经过注意力作用后的图像信息进行拼接，然后预测输出标题句子。

3.1 编码器-解码器整体框架

本课题的研究将在编码器-解码器框架下进行，通过编码器将图像信息编码，然后使用编码器的输出循环解码为一句话。本节先介绍两种编码器-解码器的框架结构，后续的改进以及对比也是基于这两种网络结构。

第一种网络结构，如图 3-1 所示，图中主要包含两部分结构：（1）编码器部分，编码器使用卷积神经网络中的 VGG 网络提取图像特征作为编码输出，VGG 网络在上一章已经介绍过，此处不再赘述，图像特征通常选择 VGG 网络中的最后一层卷积层特征或者第一层全连接层特征，对应于表 2-1 中就是网络 D 中第五部分第三个卷积层特征和第六部分的第一个全连接特征；（2）解码器部分，解码器以编码器输出的图像特征作为输入，使用 LSTM 作为解码器的算法，循环将图像解码为一个标题句子，“a man saw a dog”，其中 BOS 和 EOS 分别代表着句子的起始和结束。

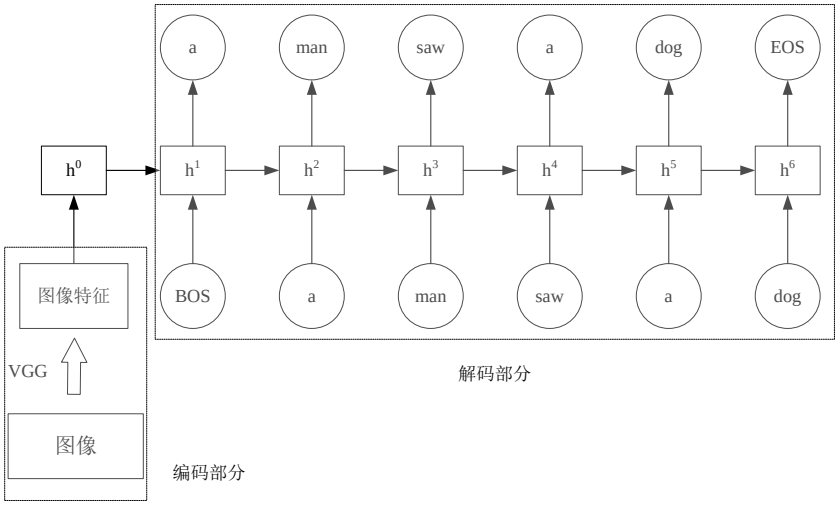


图 3-1 无注意力的 NIC 模型

第二种网络结构，如图 3-2 所示，与第一种网络同样包含编码器和解码器部分，但是不同的是，在解码的过程中加入了注意力向量α，由编码器部分生成的图像特征不仅输入到 LSTM 的初始状态中，还作为每一个时刻注意力向量的输入。

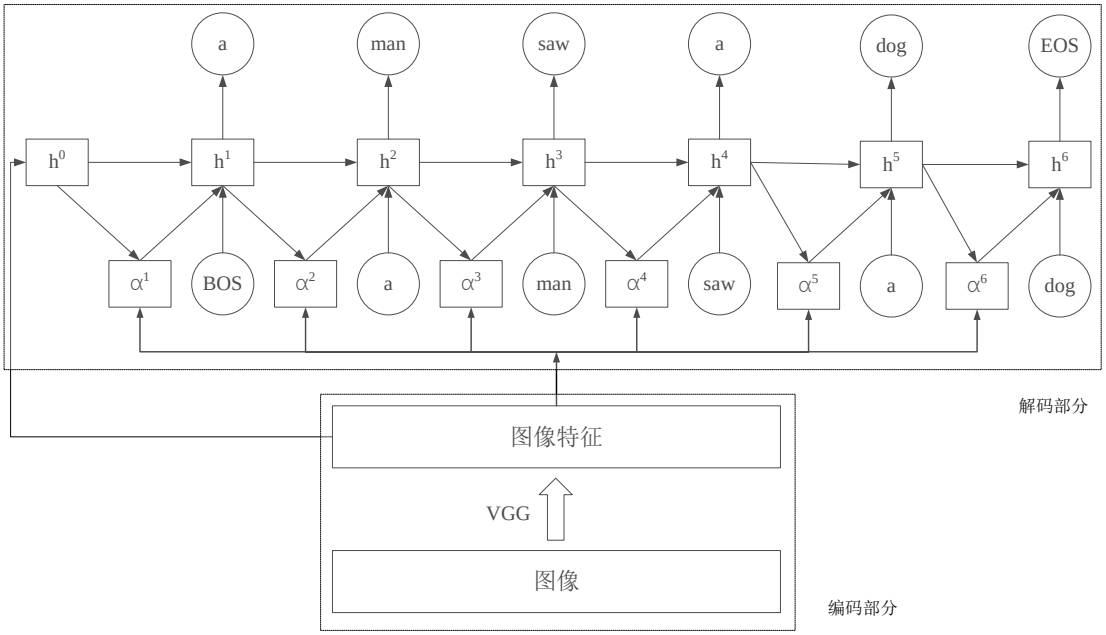


图 3-2 有注意力的 NIC 模型

本章后续小节主要对上述两种网络架构分别进行改进，根据各自结构分别提出了 past-feeding 和 past-attention 两种改进思路，并且由于第一种网络被包含在第二种网络中，所以针对第一种网络的 past-feeding 改进同样可以应用到第二种网络中，改进的思路将在后续小节中具体阐述。

3.2 基于 past-feeding 的图像标题生成算法

3.2.1 算法基本思想

解码器是整体网络结构中非常重要的一部分，通常由 LSTM 来完成标题句子的解码工作。从图 3-1 中可以看到，LSTM 的初始隐藏层状态是由图像特征经过变换所得到，以后的每一时刻的隐藏层信息都由当前时刻的输入和前一时刻的隐藏层信息所决定，因此初始隐藏层中的图像信息会影响到后续每一次的输出结果，用更加通俗的说法来描述就是，给计算机看一次图片然后让它一次性把整个标题句子写出来。理想情况下，LSTM 中每个时刻的隐藏层包含的信息不仅有图像信息，还有语法信息，因为解码输出的标题句子必须符合语法规范，例如，“a man saw a dog”这句话符合英文语法规范，而“a man saw a dog a

dog”这句话不符合语法规范，句子最后的“a dog”显然是多余的，这种情况从直觉来看，是算法并没有察觉到自己已经输出了“a dog”这个片段，如何让算法意识到自己已经输出了什么信息，是本节 past-feeding 算法主要改进思路。

为了算法知道自己已经输出了什么信息，本文提出了 past-feeding 算法改善解码输出质量，虽然 LSTM 本身自带记忆功能，但是实际仍然会出现刚才举例的情况，可见 LSTM 的隐藏层中记忆的内容以及方式并非能够像理想中情况一样，既能记住所有信息还能将信息任意转化为理想中的形式。本文使用 past-feeding 算法辅助 LSTM 共同决策当前时刻的输出，如图 3-3 所示，为了让图更加简洁，省去了编码器部分。从图中可以发现，past-feeding 向量是由每个时刻的输入联合组成，并且每个时刻包含的信息内容不一样。假设当前时刻为 t，则此时 past-feeding 向量中包含了从 1 到 t 时刻的词向量信息，预测 t 时刻的输出时，联合了 t 时刻的隐藏层和 past-feeding 向量一起做出决策。具体 past-feeding 向量的计算可以使用词向量求和的方式，可以根据问题的需要，自行定义求和长度。

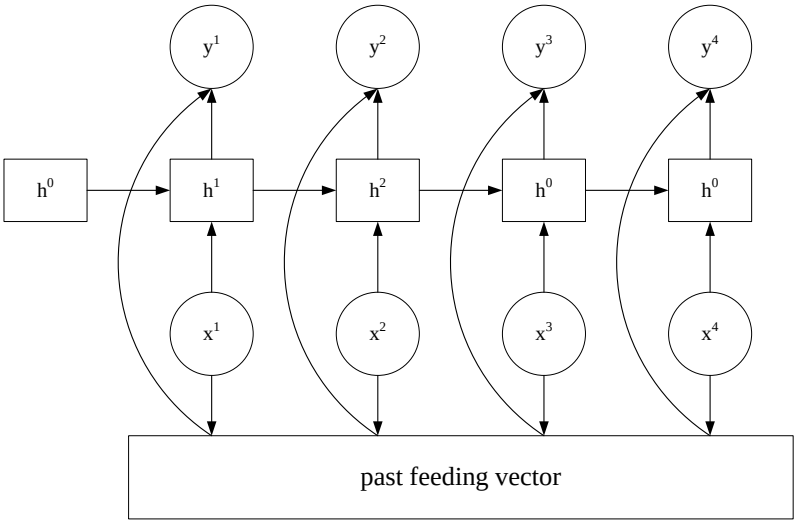


图 3-3 基于 past-feeding 的图像标题生成算法

3.2.2 公式推导

通过 3.2.1 节对 past-feeding 的算法基本原理介绍，以及第二章对深度学习基础知识的叙述，本小节将从公式推导的角度进一步阐述 past-feeding 算法。公式推导的内容包含了算法的所有内容，不过为了使公式更加简洁明了，推导过程会使用第二章所定义的公式包，具体公式包的细节在第二章已经详细推导过，此处不再展开赘述。下面是具体的公式推导过程，

$$x_{feat} = VGG(x_{img}) \tag{3-1}$$

$$c^{(0)} = W_{c0} \cdot x_{feat} + b_{c0} \tag{3-2}$$

$$h^{(0)} = W_{h0} \cdot x_{feat} + b_{h0} \tag{3-3}$$

$$c^{(t)}, h^{(t)} = LSTM(x^{(t)}, c^{(t-1)}, h^{(t-1)}) \tag{3-4}$$

$$pf^{(t)} = \sum_{i=1}^{t} x^{(i)} \tag{3-5}$$

$$y^{(t)} = soft\max(W_{h2y} \cdot h^{(t)} + W_{pf2y} \cdot pf^{(t)} + b_y) \tag{3-6}$$

公式（3-1）是一个公式包，代表将原始图像 x_{img} 输入 VGG 网络后得到图像特征 x_{feat}，具体 VGG 网络所使用的卷积、池化操作的公式以及网络结构已经在第二章叙述过。当得到图像特征后，对图像特征进行线性变换后作为 LSTM 的初始状态，具体如公式（3-2）和（3-3）所示，W_{c0} 和 b_{c0} 是得到初始状态 c 的线性变换的参数，W_{h0} 和 b_{h0} 是得到初始状态 h 的线性变换的参数。公式（3-4）是解码器所使用的 LSTM 算法的公式包，除了初始状态以外，其他的每个时刻的状态都可以由前一个时刻的状态计算得到，而初始状态则通过公式（3-2）和（3-3）得到。公式（3-5）是 past-feeding 向量的计算公式，可以看到，其本质就是对输入的多个词向量求和，在 t 时刻时，past-feeding 向量由 1 到 t 时刻的输入词向量 x 求和得到，当然，根据问题的不同，可以调整求和长度。

3.3 基于 past-attention 的图像标题生成算法

3.3.1 算法基本思想

注意力模型是图像标题生成算法中重要的一部分，让解码过程中输出的每一个单词都对应到图像中的某一部分，在每次预测时关注的图像内容越少，被其他无用信息干扰的可能性就越少，进而提升解码质量。本课题所使用的注意力算法是软注意力算法，即解码的每个时刻都会生成一个注意力向量，这个向量中不同位置的实数值代表算法对图像中不同部分的重视程度，越接近于 1 代表越关注，接近 0 则代表不关注，例如，当输出单词“man”时，图像的特征图中人的区域的注意力向量的值应该接近于 1，而其他背景区域的注意力向量的值接近于 0。目前的软注意力算法的输入包括图像信息和前一时刻 LSTM 的隐藏层信息，前后两个时刻的注意力向量并没有直接建立联系，本文提出 past-attention 算法，将每个时刻的注意力向量之间使用 LSTM 建立联系，让算法记住已经生成过图像中哪些部分的注意力，根据图像中已经关注过的内容生成新的注意力向量。例如，“a man saw a dog”，当注意力已经在图像中“man”的部分停留过后，后续注意力不应该继续在该部分停留，应该将注意力转移到图像的其他部分。此外，本文提出的 past-attention 还将解码器中的整体架构进

行修改，将其分成两个部分，语言信息部分和图像信息部分，使整体结构更加清晰。值得注意的是，past-attention 和上一小节的 past-feeding 二者都是为了更好的使用历史信息，而 past-attention 是针对图像信息部分，past-feeding 是针对语言信息部分，二者可以同时并行使用。

如图 3-4，整体网络结构依然包含两个部分，编码器和解码器，不一样的是解码器中还包含了两部分，语言信息部分和图像信息部分，语言信息部分的输入和图像无关，不过图像信息部分的输入与语言信息相关，因为只有知道当前句子的前文信息，才能决定当前时刻对图像的注意力在哪。语言信息部分的建模依然使用 LSTM，而图像信息部分中的注意力向量计算方式和原始的注意力模型并不一样，原始的注意力模型如图 2-7 所示，每个时刻的注意力向量只有 2 个输入。本文提出的 past-attention 算法对注意力向量的计算方式进行了改进，不仅包含了图像信息和前一时刻的 LSTM 的隐藏层信息，还包含了前一时刻的注意力向量信息，也就是说，每个时刻的注意力向量有 3 个输入。这种将前后的注意力向量关联起来的方式，有助于在历史的注意力的基础上，生成当前时刻的注意力向量。与语言信息建模一样，本文使用 LSTM 算法对前后注意力向量的联系进行建模。当语言部分 LSTM 和图像注意力部分 LSTM 的隐藏层都计算完毕后，使用二者的信息联合预测当前时刻的输出。整体算法结构相比原始的注意力模型更加清晰，不同部分负责不同的功能，而不是将语言和图像信息同时输入到一个 LSTM 中。

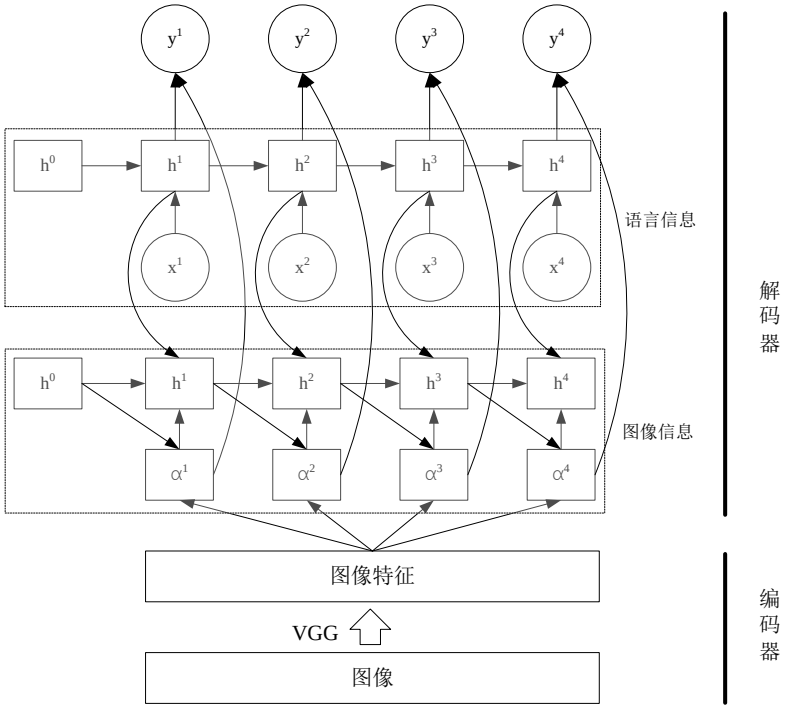


图 3-4 基于 past-attention 的图像标题生成算法

3.3.2 公式推导

通过上一小节对 past-attention 算法基本思想的介绍，本节对算法的公式进行推导。与 past-feeding 的公式推导一样，为了保证完整性，本节推导内容将包括算法中的所有部分，同时为了保证简洁性，将使用公式包代表一些特定算法的公式集合，但是本节的公式推导中涉及了两个 LSTM 部分，为了防止混淆，将使用下标进行区别。由于本节的公式相比于之前的章节更多，所以对于每个部分的公式分开阐述。

$$x_{feat} = VGG(x_{img}) \tag{3-7}$$

与 3.2.2 节中一样，公式（3-7）一样作为图像特征抽取部分，也就是编码器部分，但是不一样的是，一般来说 3.2 节中描述的网络抽取 VGG 网络中第一层全相连特征，特征维度为 4096，而此处抽取的图像特征为 VGG 网络中的最后一层卷积层特征，特征维度为 512*196，其中 512 代表特征图的个数，196 代表特征图的大小，抽取出的图像特征将用于后续的计算当中。

$$c_{txt}^{(t)}, h_{txt}^{(t)} = LSTM_{txt}(x^{(t)}, c_{txt}^{(t-1)}, h_{txt}^{(t-1)}) \tag{3-8}$$

公式（3-8）为解码器中的语言信息部分的公式，从公式中可以看到，建模语言信息使用 LSTM 算法，其中 $LSTM_{txt}$ 代表建模语言信息的 LSTM，后面简称为文本 LSTM，$c_{txt}^{(t)}$ 代表 t 时刻文本 LSTM 的记忆存储器，$h_{txt}^{(t)}$ 代表 t 时刻文本 LSTM 的隐藏层信息，值得注意的是，文本 LSTM 中的初始状态并没有给出计算公式，可以直接初始化为零。

$$\alpha^{(t)} = ATT(a, h_{att}^{(t-1)}) \tag{3-9}$$

$$z^{(t)} = \sum_{i=1}^{L} \alpha_i^{(t)} \cdot a_i \tag{3-10}$$

公式（3-9）为计算注意力向量的公式，同样使用第二章介绍注意力模型时定义的公式包，$h_{att}^{(t-1)}$ 代表前一个时刻图像 LSTM 的隐藏层信息。公式（3-10）为计算经过注意力向量作用后的图像信息，在第二章已经介绍过。

$$c_{att}^{(0)} = W_{ca0} \cdot x_{feat} + b_{ca0} \tag{3-11}$$

$$h_{att}^{(0)} = W_{ha0} \cdot x_{feat} + b_{ha0} \tag{3-12}$$

$$c_{att}^{(t)}, h_{att}^{(t)} = LSTM_{att}(\alpha^{(t)}, h_{txt}^{(t)}, c_{att}^{(t-1)}, h_{att}^{(t-1)}) \tag{3-13}$$

公式（3-9）至（3-13）共同组成了解码器中的图像信息部分，其中 $LSTM_{att}$ 代表建模图像信息的 LSTM，后面简称为图像 LSTM。公式（3-11）和（3-12）是生成图像 LSTM 的初始状态的公式，和文本 LSTM 不同，图像 LSTM 使用图像特征计算出初始状态，而文本 LSTM 直接将初始状态赋值为零。公式（3-13）是图像 LSTM 的公式包，需要注意的是，图像 LSTM 的输入参数有 4 个：（1）

当前时刻的注意力向量；（2）当前时刻文本 LSTM 的隐藏层信息；（3）前一时刻图像 LSTM 的记忆存储器信息；（4）前一时刻图像 LSTM 的隐藏层信息。为了兼容第二章介绍 LSTM 时定义的 3 个参数，可以将前 2 个参数拼接成 1 个。

$$y^{(t)} = soft\max(W_{ht2y} \cdot h_{txt}^{(t)} + W_{z2y} \cdot z^{(t)} + b_y) \tag{3-14}$$

公式（3-14）是最后输出层的计算公式，从公式中可以看到，最终的输出由当前时刻文本信息和图像信息共同计算得到，$soft\max$ 函数将结果归一化。

3.4 实验结果分析与可视化

通过本章前 3 节的介绍，大致了解了本文对图像标题生成任务的算法设计以及相应的改进思路，接下来，本节将对之前所描述的网络结构分别进行实验，并对实验结果进行分析以及可视化。

3.4.1 实验环境、数据与整体流程

由于运行深度学习算法需要消耗大量计算资源，使用 CPU 运行显然不可行，只有使用并行计算能力大大强于 CPU 的 GPU 设备才能支持深度学习算法快速的计算，根据 GPU 设备的好坏，加速比通常能够达到几十倍甚至上百倍。本课题将使用 6G 显存的 Titan Black 显卡作为并行计算设备，所有实验均在 Intel® Core i7-2600 CPU @3.4GHz 和 32G 内存的 64 位 Ubuntu 14.04 系统中完成，所有代码均使用 python 语言编写，基于 mxnet 构建整体网络结构以及完成训练过程。mxnet 是一个深度学习框架，相比于其他的 theano、tensorflow、caffe、torch 等深度学习框架，mxnet 有着大量的优点，例如支持多种语言、支持多机多卡训练、自动求导、节省显存、运行效率高等。

表 3-1 数据集

数据集	图片数	标题数	验证集大小
Flickr8k	8000	40000	1000
Flickr30k	30000	150000	1000
Coco	120000	600000	5000

如表 3-1 所示，本课题的实验数据主要包含 3 个，Flickr8k、Flickr30k 和 Coco 数据集，这 3 个数据集均为公开数据集，可以从互联网中下载得到。Flickr8k 数据集包含 8000 图片，每张图片有 5 个标题句子与之对应，共 40000 条标题。同理，Flickr30k 数据集包含约 30000 张图片，150000 条标题，Coco 数据集包含约 12 万张图片，600000 条标题。实验开始前先划分数据集，从 3 个数据集

中分别抽取出 1000、1000、5000 张图片和其对应的标题句子当做验证集，其余的当做训练集，训练集用于训练模型参数，验证集用于评价模型的好坏以及判断是否过拟合使用，当模型过拟合时应使用早停止策略，停止训练并保存模型。这 3 个数据集中的图像均为真实场景图像，没有经过人为的拼接、修饰等操作，也不是漫画等非真实场景图像。如图 3-5 所示，图中展示了 9 张数据集中的样例图片，每张图片都是拍摄于真实场景。

图 3-5 真实场景图片

本课题的实验整体的运行流程如图 3-6 所示，实验的输入包含图像和对应的标题，整体流程的第一步是数据预处理，包含图像预处理和文本预处理，图像预处理主要指将图像压缩成指定大小、图像去均值化、将少数单通道图转化为三通道图等操作，文本预处理主要指将文本转化为指定长度的向量，长度超过的进行截断，长度不足的进行填充，并且文本中的每个单词以数字表示。图像数据处理完后使用 VGG 网络提取特征，将提取的特征与预处理过后的文本一起存入 HDF5 格式文件中，模型训练过程中会循环读取 HDF5 文件中的内容，供算法运行使用，模型每次迭代都会输出验证集的效果，根据验证集的效果判

断当前模型是否过拟合，如果未过拟合，则继续训练，反之则直接保存最优的那一次模型，并输出验证集结果。

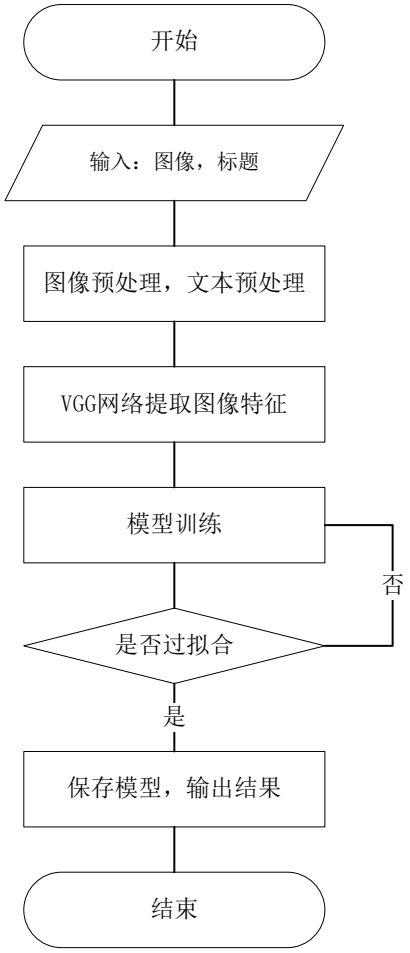


图 3-6 实验整体流程

3.4.2 评价指标

由于图像标题生成算法最终输出的是一个句子，需要将生成的标题句子和正确的标题句子进行相似度比较，所以无法用传统的准确率、精度和召回率等指标进行评价，而使用人工的方式对句子的相似度进行评价又太过昂贵，并且无法复用，本课题将使用 3 种常用的图像标题生成算法的评价指标 BLEU[51]、ROUGE[52]、CIDER[53]，来对生成的图像标题句子进行评价。其中，BLEU 评价指标是以待评价句子在参考句子中 n 元词片段命中精度为基础进行计算，根据 n 的取值不同，又可以分为 BLEU1、BLEU2、BLEU3、BLEU4 四种，ROUGE 评价指标是以待评价句子和参考句子的最短公共子序列长度为基础进行计算，CIDER 评价指标是以待评价句子和参考句子的 TFIDF 向量的相似度为基础进行计算。为了节省篇幅，本节只对 BLEU 算法进行简单介绍。

针对于本课题，生成什么样的图像标题是好的标题？答案是越接近于人类对图像标题的描述方法越好。因此，设计一个好的评价指标要尽量符合人类对图像的描述习惯，但是这本身很困难。本文使用的 BLEU 算法提供了一种途径，以数值的形式量化生成标题的好坏，其思想与语音识别中的词错误率（WER）相类似。通常，BLEU 算法根据词片段的长度不同分为 N 个部分，BLEU1 代表以一元词片段作为计算基础，以此类推，一般 N 取值为 4。

$$BLEU = BP \cdot \exp(\sum_{n=1}^{N} w_n \log p_n) \tag{3-15}$$

公式（3-15）是 BLEU 评价指标的计算公式，其中 BP 为语句简短惩罚因子，为了防止语句过于简短而导致的结果好的假象，p_n 表示以 n 元词片段为基础的修正精度，w_n 为相应的权重，如果没有权重的先验知识，可以直接设置为均匀分布，即 $1/N$。

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{ngram \in C} Count_{clip}(ngram)}{\sum_{C' \in \{Candidates\}} \sum_{ngram' \in C'} Count(ngram')} \tag{3-16}$$

$$Count_{clip} = \min(Count, Max\_ref\_Count) \tag{3-17}$$

公式（3-16）是以 n 元词片段为基础的修正精度的计算公式，其中 $Count_{clip}$ 由公式（3-17）可得，表示分别统计词片段在待评价句子和参考句子中出现的次数，并取较小值，例如，“a dog”是一个二元词片段，在待评价句子中出现 3 次，参考句子中出现 2 次，则此时 $Count_{clip}("a\ dog")$ 的值为 2。

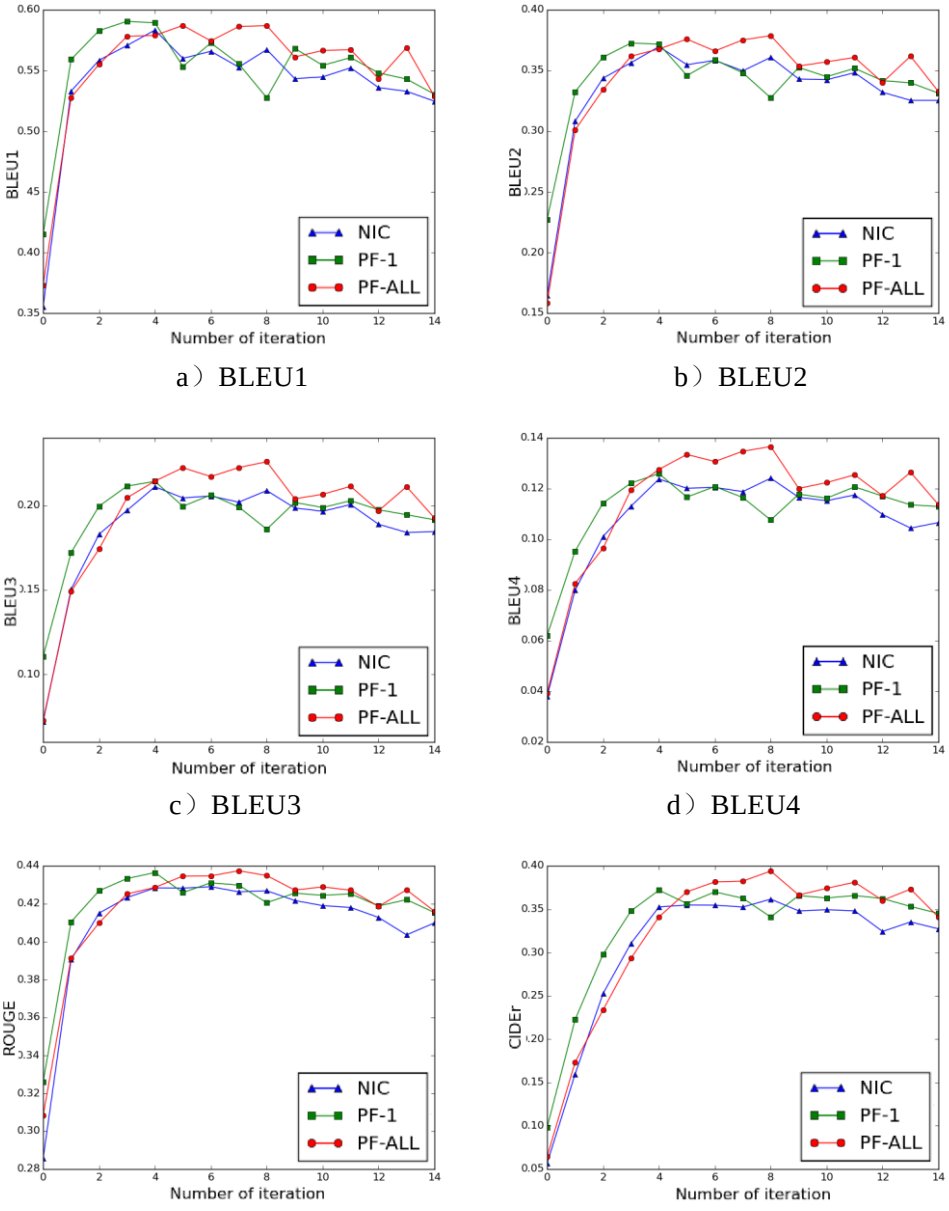
$$BP = \begin{cases} 1 & c > r \\ e^{(1-r/c)} & c \le r \end{cases} \tag{3-18}$$

公式（3-18）是语句简短惩罚因子的计算公式，其中 c 代表待评价句子的长度，r 代表参考句子的长度，当参考句子有多个时，取最接近待评价句子长度的参考句子的长度，即最佳匹配长度，作为 r 的值。

3.4.3 实验结果与分析

本章的实验对比算法是 NIC 和 ATT-NIC，二者都是基于编码器-解码器框架，后者相对于前者多了注意力模型。本章将对 5 个算法进行实验，分别为 PF-1、PF-ALL、PA、PA+PF-1 和 PA+PF-ALL。其中，PF-1 代表使用 past-feeding 算法，并且设置计算 past-feeding 向量的求和长度为 1，即只使用当前时刻的输入当做 past-feeding 向量。PF-ALL 同样使用 past-feeding 算法，但是求和长度包含了过去所有时刻的输入。PA 代表使用 past-attention 算法，PA+PF-1 表说同时使用 PF-1 和 PA 算法，PA+PF-ALL 表示同时使用 PF-ALL 和 PA 算法。为了保证实

验的公平性，本章的所有实验使用的超参数保持一致，样本批次大小为 150，LSTM 的堆叠层数为 1，任何中间隐藏层的神经元个数为 512，词向量维度为 512，dropout 比率为 0.5，优化方法为 Adam，学习率为 0.0001，梯度剪裁阈值为 5。



a）BLEU1　　b）BLEU2

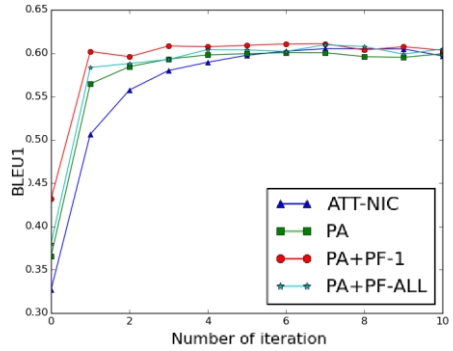
c）BLEU3　　d）BLEU4

e）ROUGE　　f）CIDEr

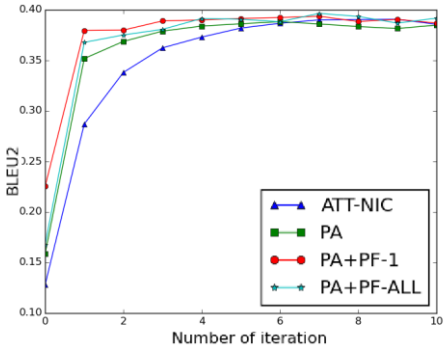
图 3-7 Flickr8k 数据集上 3 个算法在不同评价指标中的对比实验

如图 3-7 所示，为 Flickr8k 数据集上 NIC、PF-1、PF-ALL 三个算法在六个评价指标上的对比实验，其中每个图的横坐标为迭代次数，纵坐标为相应的评价指标，每个图中包含三条折线，分别代表了三个算法随着迭代次数的增多，

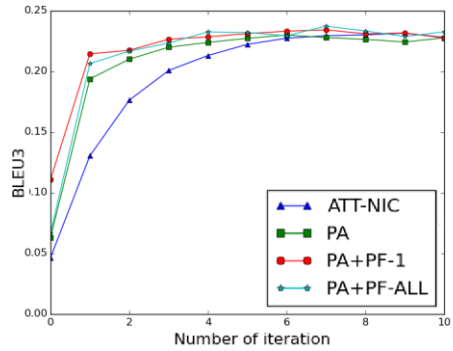
在不同评价指标中发生的变化。三角形的折线代表了 NIC 算法，正方形的折线代表了 PF-1 算法，圆形的折线代表的 PF-ALL 算法。三种算法的迭代次数均为 150 次，但是为了避免图过于混乱，将所有的结果以 10 为步长求和取平均，每个评价指标下得到 15 个值。观察图 3-7 可以发现，PF-1 算法相对其他的两个算法在迭代的初期效果提升的较快，而 PF-ALL 算法在整个迭代过程中效果最好。在评价指标 BLEU1 中，PF-ALL 算法在约 40 次迭代以后，优势逐渐明显，并且随着 BLEU 的 n 元词片段数加大，PF-ALL 算法的优势更为明显。



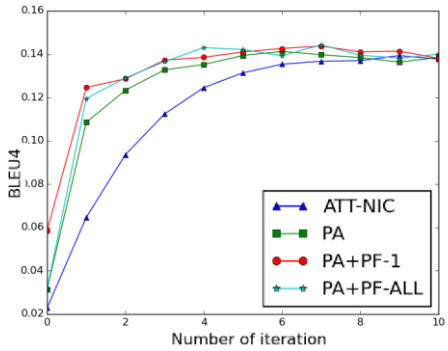
a）BLEU1



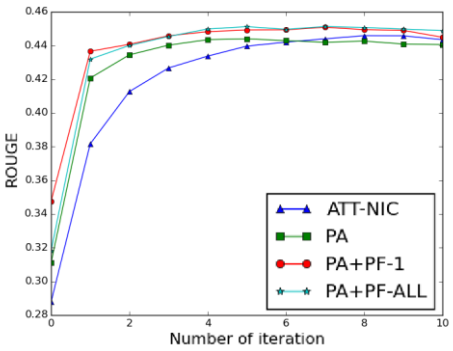
b）BLEU2



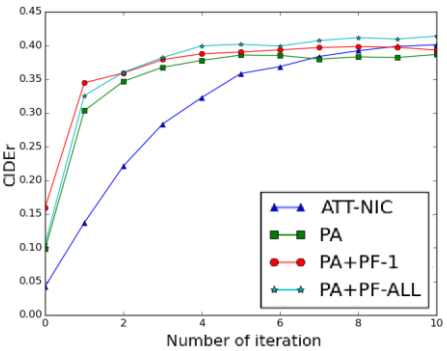
c）BLEU3



d）BLEU4



e）ROUGE



f）CIDEr

图 3-8 Flickr8k 数据集上 4 个算法在不同评价指标中的对比实验

如图 3-8 所示，为 Flickr8k 数据集上 ATT-NIC、PA、PA+PF-1 和 PA+PF-ALL 四种算法在六个评价指标下的对比实验，每个图中包含 4 条折线，分别代表了 4 种算法，其中，三角形折线代表 ATT-NIC 算法，正方形折线代表 PA 算法，圆形折线代表 PA+PF-1 算法，星形折线代表 PA+PF-ALL 算法。四种算法的迭代次数为 55 次，以步长 5 为间隔对所有结果求平均，最终每种评价指标得到 11 个结果。

观察图 3-8 可以发现，所有的评价指标都显示 ATT-NIC 算法收敛的最慢，PA 算法相比于 ATT-NIC 算法收敛速度提升很多，PA+PF-ALL 相比于 PA 收敛速度进一步提升，而收敛速度最快的是 PA+PF-1 算法。在图 3-8 的子图（a）中，各个算法收敛于 60 左右，在子图（b）中，各个算法收敛于 39 左右，在子图（c）中，各个算法收敛于 23 左右，在子图（d）中，各个算法收敛于 14 左右，在子图（e）中，各个算法收敛于 44 左右，在子图（f）中，各个算法收敛于 40 左右。从效果上来看，本文提出的 PA+PF-ALL 和 PA+PF-1 在多个指标下均有不同程度的优势。

表 3-2 PAST-FEEDING 算法实验结果

数据集	模型	评价指标					
		B-1	B-2	B-3	B-4	Rouge	CIDEr
Flickr8k	NIC	59.81	38.15	21.95	12.99	43.53	37.79
	PF-1	**60.00**	38.21	22.53	13.69	43.93	38.43
	PF-ALL	59.73	**38.58**	**23.13**	**14.27**	**44.19**	**40.27**
Flickr30k	NIC	59.31	36.88	21.59	**13.13**	42.52	35.16
	PF-1	**59.89**	**36.95**	**21.59**	12.99	**42.88**	**35.75**
	PF-ALL	59.09	36.66	21.31	12.75	42.73	35.67
Coco	NIC	68.45	47.29	30.66	20.11	51.47	69.19
	PF-1	**68.63**	47.36	30.65	20.05	51.64	69.57
	PF-ALL	68.45	**47.53**	**31.07**	**20.51**	**51.71**	**69.93**

表 3-2 是 past-feeding 算法在不同评价指标下的实验结果，整个表包含 3 个部分：数据集、算法和评价指标，每个数据集运行 3 个算法，每个算法以 6 种不同的评价指标进行评价。7 种算法具体包括：NIC、PF-1 和 PF-ALL，3 个数据集包括：Flickr8k、Flickr30k 和 Coco，6 种评价指标包括：B-1、B-2、B-3、B-4、Rouge 和 CIDEr，其中 B-1 表示 BLEU1，以此类推。观察表 3-2 可以发现，在三个不同的数据集上，PF-1 和 PF-ALL 相比于 NIC 算法在多个评价指标下均有提升，并且 PF-1 更有助于提升 B-1 评价指标，而 PF-ALL 更有助于提升 B-4 评价指标，总体来看，PF-ALL 相比于 PF-1 更有优势，因为 PF-ALL 相比于 PF-1 使用了更多的前文已输出信息。由上一小节对评价准则的介绍可以得知，

ROUGE 和 CIDER 的评价计算基础相对于 BLEU 更加优秀，因此参考价值更加高。从数据集的规模而言，Coco 数据集最大，相对于 Flickr8k 和 Flickr30k 不容易过拟合，因此 Coco 数据集的实验结果参考价值相对更高。综合以上两点，表 3-2 中最具参考价值的部分是表中右下角部分，从该部分可以看到，本文提出的 PF-1 和 PF-ALL 均有良好的表现。

表 3-3 PAST-ATTENTION 算法实验结果

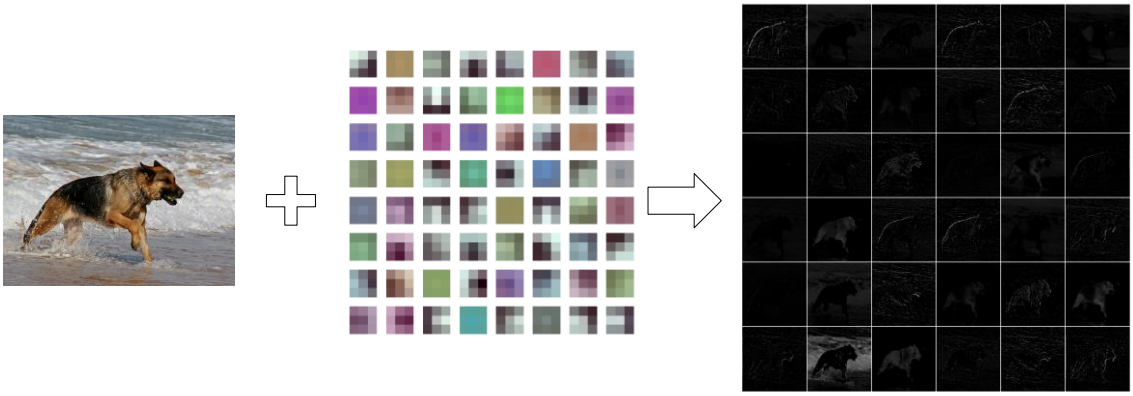
数据集	模型	评价指标					
		B-1	B-2	B-3	B-4	Rouge	CIDEr
Flickr8k	ATT-NIC	61.21	39.68	23.56	14.51	44.67	40.99
	PA	60.49	39.41	23.57	14.65	44.55	39.18
	PA+PF-1	**61.38**	39.60	23.69	**14.72**	**45.38**	40.49
	PA+PF-ALL	61.26	**39.95**	**23.91**	14.64	45.24	**42.45**
Flickr30k	ATT-NIC	61.37	**39.07**	**23.71**	**14.63**	43.65	38.74
	PA	61.55	38.49	22.82	14.15	43.77	39.10
	PA+PF-1	**61.76**	38.71	22.98	14.12	43.83	**39.59**
	PA+PF-ALL	61.50	38.66	22.93	13.99	**44.00**	39.24
Coco	ATT-NIC	**69.56**	**48.23**	**31.14**	20.14	51.56	71.25
	PA	69.26	47.68	31.08	20.56	51.82	71.51
	PA+PF-1	69.23	47.46	30.90	20.47	51.86	71.46
	PA+PF-ALL	69.12	47.55	30.99	**20.60**	**51.89**	**71.81**

表 3-3 是 past-attention 在不同评价指标下的实验结果，表的结构与表 3-2 一致，不同的是整个表包含 4 个算法：ATT-NIC、PA、PA+PF-1 和 PA+PF-ALL。观察表 3-3 可以发现，在 Flickr8k 数据集上，B-1、B-4、Rouge 指标最优的是 PA+PF-1 算法，B-2、B-3、CIDEr 指标最优的是 PA+PF-ALL 算法。在 Flickr30k 数据集上，B-1、B-4、CIDEr 指标最优的是 PA+PF-1 算法，B-2、B-3 指标最优的是 ATT-NIC 算法，Rouge 指标最优的是 PA+PF-ALL 算法。在 Coco 数据集上，B-1、B-2、B-3 指标最优的是 ATT-NIC 算法，B-4、Rouge、CIDEr 指标最优的是 PA+PF-ALL 算法。和表 3-2 的分析一样，表中最具参考价值的部分为右下角部分，从该部分可以看到，本文提出的算法相较于 ATT-NIC 算法具有良好的表现。

3.4.4 实验结果可视化

本章使用 VGG 网络提取图像特征，用作后续解码部分的输入，为了更加深入的理解 VGG 网络的工作原理以及提取的图像特征的形式，本节将可视化

VGG 网络的卷积核以及特征图，具体包含 VGG 网络 D 第一部分第一层卷积核、第一部分第一层特征图、第二部分第一层特征图和最后一层特征图。



原始图像　　卷积核　　特征图

图 3-9 图像特征提取

如图 3-9 所示，图中包含三部分，第一部分为原始图像，原始图像中描绘了一只狗在水中奔跑，第二部分为卷积核，该卷积核为表 2-1 中网络 D 的第一部分第一层卷积核，第三部分为特征图，从这些特征图中可以发现，不同的特征图刻画了原始图像的不同方面的信息，有些特征图着重捕获图中狗的纹理，另一些特征图着重捕获图中海水的纹理。

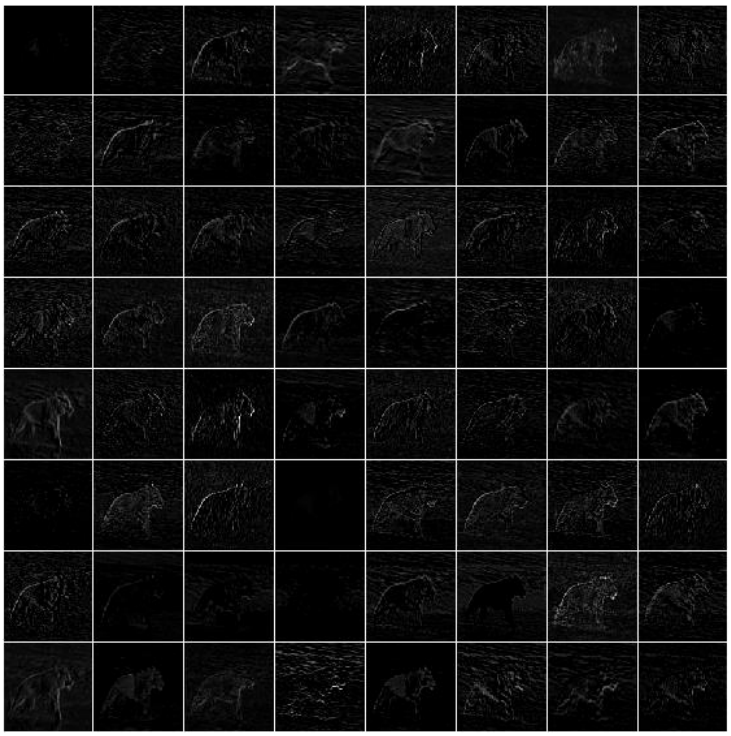


图 3-10 Conv2_1 层提取的图像特征

图 3-10 是 VGG 网络 D 的第二部分第一个卷积层的图像特征，可以看到相比于图 3-9 中的图像特征，本层的特征更加抽象，可以从图中大致看出狗的轮廓信息以及海水的纹理信息。

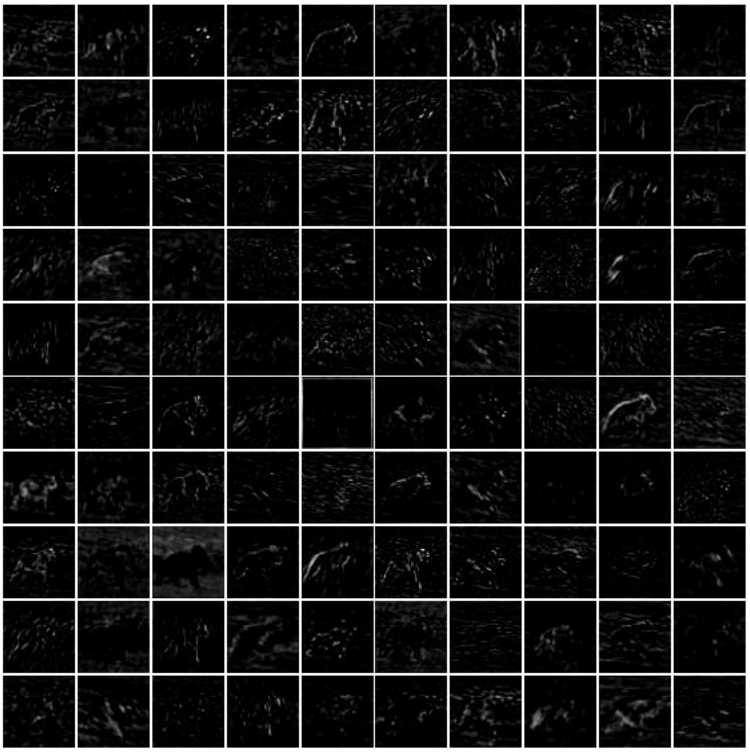


图 3-11 Conv3_3 层提取的图像特征

图 3-11 是 VGG 网络最后一个卷积层所提取的图像特征，可以看到，经过多层卷积与池化操作以后，提取的图像特征已经和原始图像有很大的差别，抽象层次非常高。

图 3-12 和 3-13 是在 Flickr8k 数据集上，使用 PA+PF-ALL 算法训练的模型生成的 2 个正确的样例，每个样例中包含了多个图片，表示了在标题句子预测过程中，每个时刻的注意力在图中的分布，其中第一张图片是基准图片，后续每一张图片都是基准图片加上注意力向量，图中白色越明显的地方，代表当前时刻的注意力更加集中的地方，而灰暗的地方，代表当前时刻不关注的信息，图片左上角为预测的单词。第一个样例的基准图像是前文提取图像特征时使用的图像，描绘了一只狗在水中奔跑，可以看到，算法正确的输出了“a dog is running through the water”，在输出单词“dog”时，注意力集中在狗的区域，输出“water”时，注意力集中在海水的区域。第二个样例描绘了一个人在划船，算法同样也正确输出了“a man is sitting on a boat in the water”，同时注意力的分布也是正确的。

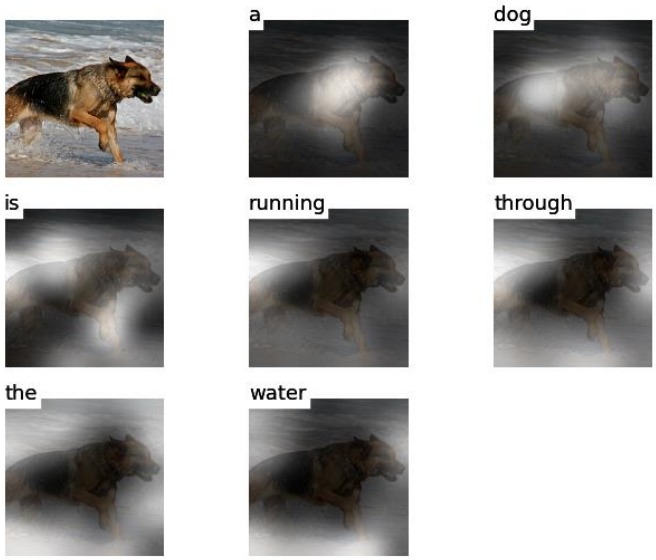


图 3-12 正确样例 A

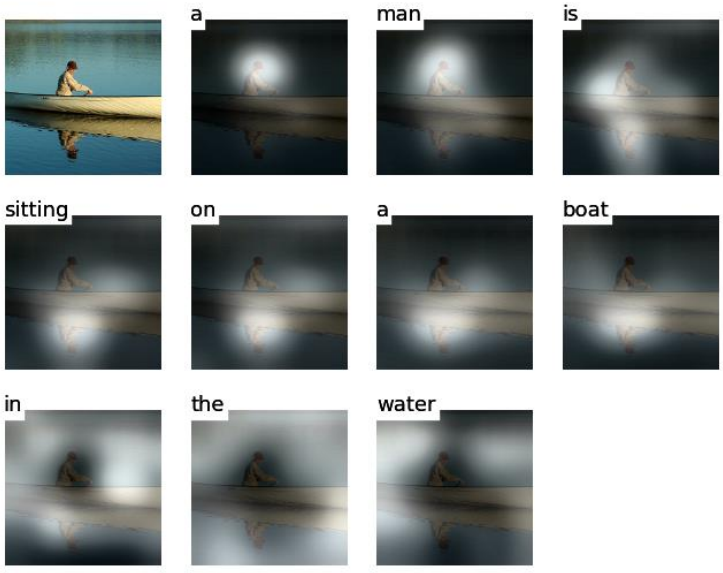


图 3-13 正确样例 B

图 3-14 和 3-15 是 2 个错误的样例。在错误样例 A 中，算法输出了“two people are sitting on a boat in a lake”，从图中可以看到并不是两个人，而是一个人和一条狗，不过令人惊讶的是，图中很难明显看到有船的部分，但是算法却判断出了“on a boat”这个概念，如果仅仅通过传统的方法识别图中的实体，不可能

输出“boat”，因为图中并没有明显出现船，这也说明了深度学习算法的有效性。在错误样例 B 中，算法输出了“two children are playing in the water”，但是实际并不是两个小孩在水中玩耍，而是两个海豚。

图 3-14 错误样例 A

图 3-15 错误样例 B

3.5 本章小结

本章主要介绍了对图像标题生成算法的研究，提出了两种针对不同网络结构的改进算法，past-feeding 和 past-attention。本章首先阐述了图像标题生成算法的整体框架，简单的介绍了两种基础算法 NIC 和 ATT-NIC。第二节介绍了本章针对于第一种 NIC 网络结构的改进算法，通过引入 past-feeding 向量辅助 LSTM 预测当前时刻的输出，具体从网络结构和公式推导两个方面进行描述。第三节介绍了本章针对于第二种 ATT-NIC 网络结构的改进算法，通过对前后时刻的注意力向量建立联系，改进注意力随着的时刻不断的有效变化。第四节介绍了本章提出的改进算法的实验，通过在 3 种数据集、多个评价指标下的实验，可以看到本章的提出的算法在不同数据集和不同评价指标下均有不同程度的改进。综上所述，本章提出的 past-feeding 和 past-attention 对图像标题生成任务有一定的效果，在多种评价指标下的表现均较为客观。

第 4 章 验证码图像标题生成系统的设计与实现

传统的验证码识别系统需要大量人工操作，并且不同的场景下系统无法很好的迁移。传统的验证码识别系统通常需要 3 个步骤：（1）去除图像噪声和图像二值化操作；（2）将二值化后图像中的每个字符分割出来；（3）逐个对分割后的字符进行识别。可以看到，第 1 步和第 2 步依赖于人工设计，并且当验证码字符个数发生变化时，整个验证码识别系统就需要重新设计。

值得注意的是，虽然传统的验证码识别是属于图像识别的问题，但是如果将验证码图片中的字符文本当做验证码图片的标题的话，验证码识别问题就可以转化为验证码图像标题生成问题。并且相比于传统的方法，使用图像标题生成算法解决验证码识别问题有诸多好处，例如不需要人工干涉、可以适应不同长度的验证码、有较高的准确率等。

4.1 数据来源与预处理

本章没有抓取网站中的验证码作为训练数据，所有数据均由 python 的 captcha 库自动生成得到，这意味着训练数据是无限的，不会受到数据量不足的影响。python 的 captcha 库是一个可以生成图片和语音验证码的工具，输入一串字符，能生成对应的图片或者语音，如图 4-1 所示，图中有 16 张 captcha 库自动生成的验证码，长度为 1~8 不等，验证码图片包含了 0~9 数字字符，每个字符非整齐摆放，颜色种类和深浅不同，包含噪声点，并且有一条贯穿所有字符的干扰线。

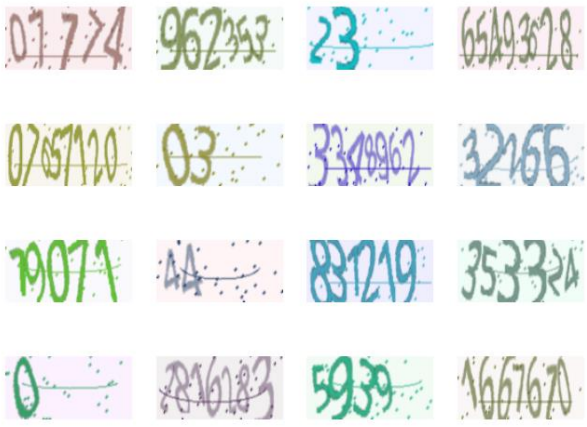


图 4-1 验证码数据集

有了验证码数据后，需要进行简单的预处理，方便后续操作使用。预处理过程主要包含 2 个，重新缩放图片为 68*116 尺寸、将 RGB 值归一化。数据预处理过后，将数据和对应的类标整理打包存入 HDF5 文件中，供数据迭代器使用。

4.2 系统整体设计

本节将详细阐述验证码标题生成系统的整体设计思路，具体包含 3 个部分，算法设计、系统实现和模型训练。算法设计中介绍了针对验证码图片而提出的图像标题生成算法，系统实现中介绍了整个系统所包含的多个模块以及模块间的关系，模型训练中介绍了本章使用的一种较优的梯度下降算法，并且给出了详细的推导。

4.2.1 算法基本思想

验证码标题生成算法的整体框架和第三章所叙述的大体一致，都服从于编码器-解码器框架，不过由于验证码图片的特殊性，在算法细节处有所变化。主要的变化包含两个方面：（1）验证码图片相比于真实场景图片，图片的尺寸更加小，因此算法的编码器部分不需要使用 VGG 网络提取图像特征，可以使用更加简单的卷积神经网络提取验证码图片特征，这么做的好处是减少算法的计算量，提高运行效率；（2）通常图像标题句子中的前后两个单词之间是高度相关的，因为有语法的存在，而验证码的每个字符之间没有任何关系，纯粹是随机生成的，因此模型中不需要对文本信息进行建模。

基于以上对验证码标题生成问题的讨论，本文针对性的提出 OCR-IC 算法。如图 4-2 所示，OCR-IC 算法同样包含 2 个部分，编码器和解码器，相比图 3-4 中的网络结构，编码器中不再使用 VGG 网络，而是使用一个自定义的更加简单的卷积神经网络，该卷积神经网络包含 3 个卷积层，每个卷积层输出 32 个特征图，并且跟着 2*2 的池化层和 ReLU 非线性激活层，前 2 个卷积层使用 5*5 的卷积核，第 3 个卷积层使用 3*3 的卷积核，由于考虑到运行效率，没有全部使用 3*3 的卷积核。本章使用的验证码图片大小为 68*116，经过 3 层卷积池化后，最终生成 32 个 6*12 的特征图。此外，OCR-IC 算法的解码器只包含图像信息部分，并不包含语言信息部分，因此图 4-2 中只有一条 LSTM 用于建模图像信息，值得注意的时，最终预测输出时只依赖于经过注意力向量作用后的图像信息，从图像 LSTM 的隐藏层中并没有信息流入输出层。

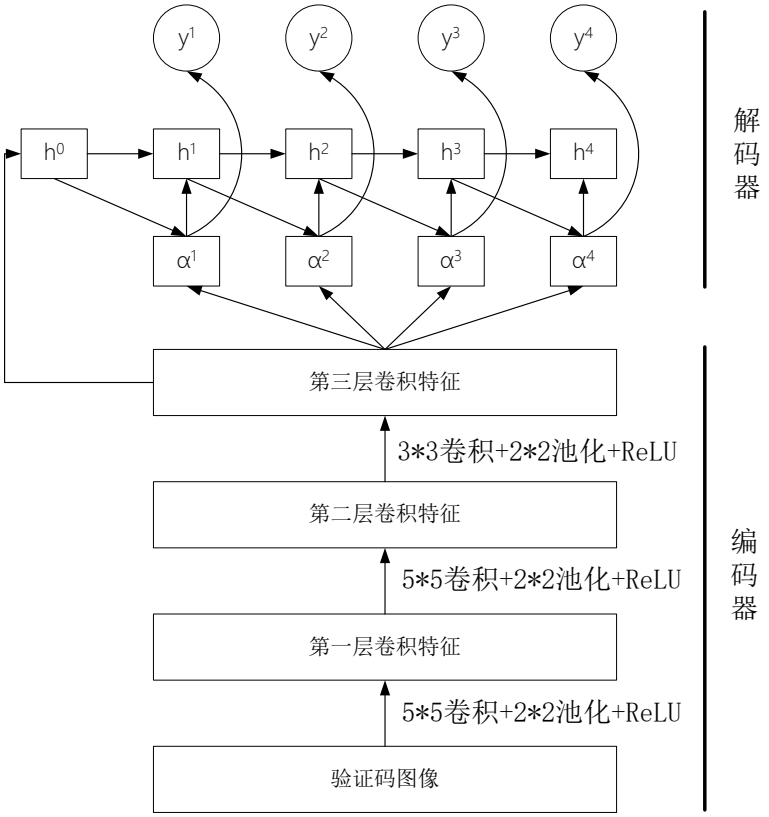


图 4-2 OCR-IC 算法网络结构

4.2.2 系统实现细节

通过 4.2.1 节对算法的基本原理的介绍，本节将从系统实现的角度进一步说明验证码图像标题生成系统的搭建及运行流程。本章所实现的验证码标题生成系统依然使用 python 作为编程语言，mxnet 作为深度学习框架，Titan Black 显卡作为并行计算资源。在系统实现过程中，并没有使用 mxnet 中已经封装好的模型训练 FeedForward 方法，而是完全自己实现了一套包含数据预处理、数据加载、模型训练、模型保存、模型评价和结果可视化等多个模块的系统，这样做的好处是对整个系统的掌控力更加强，使用更加灵活。

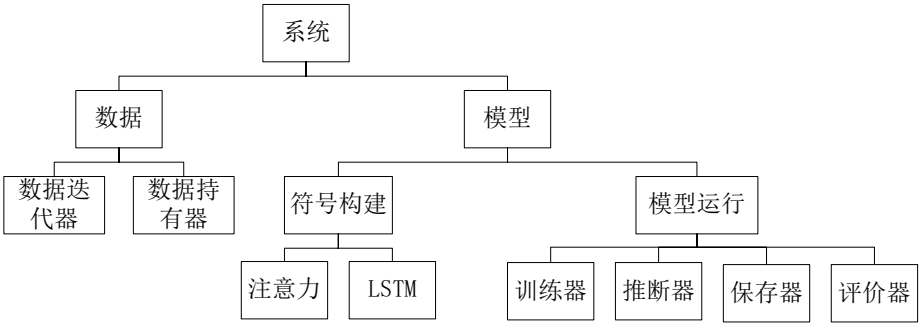


图 4-3 系统设计模块

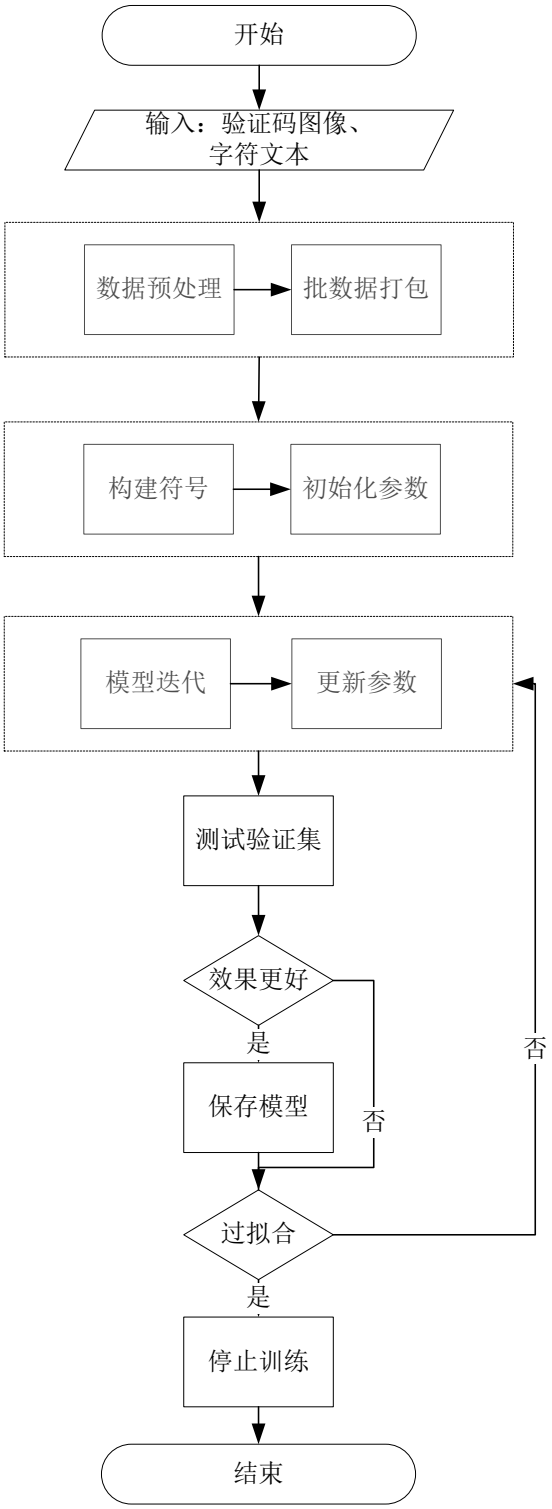


图 4-4 系统流程图

如图 4-3 所示，整个系统分数据和模型两大部分。数据部分包含数据迭代器和数据持有器，由于本章对于验证码图片的预处理步骤并不复杂，所以直接

将数据预处理包含入数据迭代器中，数据迭代器用于处理数据为可用格式并提供给后续步骤使用，数据持有器用于将一批数据打包，后续步骤只需从数据持有器中拿出数据即可。模型部分包含符号构建和模型运行，符号构建是指使用 mxnet 提供的基础操作单元构建出算法所需的符号计算图，有了符号计算图等价于有了 4.2.1 所描述的网络结构，模型运行部分又包含 4 个部分，训练器、推断器、保存器和评价器。训练器主要负责的工作是模型参数初始化、模型训练迭代、训练日志保存等，推断器主要负责新的数据来临时，模型的预测工作，保存器主要负责保存历史模型训练结果最优的模型参数，评价器负责模型的评价工作，实现了多种评价准则。

图 4-4 是系统整体流程图，系统第一步先进行数据预处理，将预处理好的一批数据进行打包，接着构建符号计算图，根据符号计算图创建相应的执行器并初始化模型参数，接下来开始训练模型，每次迭代训练包含 2 个过程，前向和后向，前向产生输出，后向生成梯度，根据生成的梯度以某种梯度下降算法的策略更新模型参数，本文所使用的梯度下降算法为 Adam，参数更新完毕后，在新的模型参数下测试验证集，得到当前参数下验证集的效果，如果效果比历史最好的效果更好，则保存当前的模型参数，接着判断模型当前是否过拟合，如果过拟合则直接停止训练模型，否则继续模型的下一次迭代。图中的虚线框是为了避免流程图过于长，而将同一组操作放在一个虚线框中，框内流程的执行顺序为从左到右。

4.2.3 模型训练方法

目前深度学习算法参数的求解主要是基于梯度下降方式，原始的梯度下降主要分为两种，批梯度下降和随机梯度下降。批梯度下降是指每次参数更新时使用所有样本产生的梯度的平均值用作参数更新，这种方式优点是能够使损失函数总是朝着下降的方向变化，缺点是参数更新太慢，每次参数更新都需要计算完所有样本的梯度后求平均值，非常浪费计算资源。随机梯度下降是指每次参数更新都只需要计算一个样本的梯度值，这种方式优点是大大加快的参数更新速度，缺点是并不能保证损失函数总是朝着下降的方向变化，并且损失函数下降到一定程度时开始震荡。多数深度学习算法使用上面两种折中的办法，即迷你批梯度下降，每次参数更新时使用小批量样本的梯度平均值，在很多文献中将迷你批梯度下降归入随机梯度下降法，对二者并不区分，本节后续内容中的随机梯度下降均指迷你批梯度下降。

虽然随机梯度下降可以作为参数求解的算法，但是并不能保证能很快的收敛到很优的极值点，原始的随机梯度下降包含以下一些问题：

1) 很难选择一个合适学习率，学习率太小会导致收敛时间非常长，学习率太大则会导致损失函数不收敛甚至发散。
2) 使用一些学习率衰减策略可以有助于训练，例如退火策略，使损失函数下降到一定阈值时，自动减小学习率，但是衰减策略以及阈值的选择都和具体的机器学习任务相关，无法很好的通用。
3) 深度学习是一个高度非凸的优化问题，损失函数有许多局部极小值点，如何绕过局部极小值点，收敛到全局最优点，是一个重要的需要解决的问题。

为了解决以上问题，近几年出现了多种随机梯度下降的变种算法，例如基于动量的随机梯度下降[54]、Adagrad[55]、Adadelta[56]、RMSProp 和 Adam[57]等。本文将使用 Adam 算法作为模型参数求解算法，Adam 算法是一种学习率自适应的随机梯度下降算法，不需要人为指定学习率衰减策略，并且相比于原始随机梯度下降，Adam 的损失函数的下降速度更快。下面是 Adam 算法的公式推导，

$$m_t = \beta_1 m_{t-1} + (1-\beta_1) g_t \tag{4-1}$$

$$v_t = \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \tag{4-2}$$

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \tag{4-3}$$

$$\hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{4-4}$$

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{\hat{v}_t} + \varepsilon} \cdot \hat{m}_t \tag{4-5}$$

公式（4-1）中变量 m_t 用于保存梯度的历史信息，类似于动量的效果。公式（4-2）中变量 v_t 用于保存梯度平方的历史信息，主要用于自适应的改变学习率。当使用零向量初始化变量 m_t 和 v_t 时，会导致后续的值更偏向于 0，尤其是当 β_1 和 β_2 接近于 1 时，因此使用公式（4-3）和（4-4）对二者进行校正。公式（4-5）是参数更新公式，使用校正过后的 $\hat{m}_t$ 和 $\hat{v}_t$ 计算得到，其中 η 为学习率，ε 是一个很小的数，为了防止分母为 0 而导致的计算异常。

4.3 实验结果分析与可视化

通过 4.1 和 4.2 节对数据和算法的叙述，本章将对 OCR-IC 算法进行实验，在正式进行实验之前，会先简单介绍对比算法，后续实验结果也将包含对比算法的运行结果，最终根据实验结果分析本章提出的 OCR-IC 算法的优势，并进行实验结果可视化工作。

4.3.1 对比算法

本章所包含的对比算法主要包含 2 个，CNN 和 LSTM+CTC。二者各有优劣，第一种 CNN 方法是指单纯使用 CNN 提取验证码图像特征后直接进行分类，假设验证码中有 4 个字符，则使用图像特征进行 4 次分类，每个分类器对应一个位置的验证码字符，顺序不能乱。第二种方法是对验证码图像逐列像素输入到 LSTM 中，对每列像素进行分类，使用 CTC 损失函数，这种做法和语音识别非常相似。

4.3.2 评价指标

本章算法的评价指标包含 2 个，单个字符的准确率和全字符的准确率。单个字符的准确率是以字符为单位的准确率，假设预测输出的验证码字符为“3376”，真实的验证码字符为“3316”，则单个字符的准确率为 0.75。全字符的准确率是以句子为单位的准确率，只有全部字符都正确才算预测正确，否则为预测错误。

4.3.3 实验结果与分析

本章将对 5 个算法进行实验，具体包括 CNN、LSTM+CTC、OCR-IC4D、OCR-IC4B 和 OCR-IC8B，其中 CNN 和 LSTM+CTC 算法的字符为定长，长度为 4，OCR-IC4D 算法的字符为定长，长度为 4，OCR-IC4B 算法的字符为变长，长度为 4，OCR-IC8B 算法的字符为变长，长度为 8。本章所有实验的参数保持一致，LSTM 的隐藏层神经元个数为 100，LSTM 堆叠层数为 1，dropout 率为 0.5，学习率为 0.001，梯度剪裁阈值为 5，包含 3 个卷积层，卷积核大小分别为 5、5、3，每层的特征图个数为 32，每个卷积层后跟随一个池化层和一个非线性激活层，所有池化层的核大小为 2，步长为 2，训练集大小为 10000，验证集大小为 1000，迭代次数为 500 次。

如图 4-5 所示，为多个算法的迭代收敛图，图中包含 5 条折线，分别代表 5 个算法，横坐标为迭代次数，纵坐标为全字符准确率，为了让图更加清晰，将所有的结果每 10 次迭代求平均值，得到 50 个结果，因此横坐标的长度为 50。从图中可以发现，CNN、LSTM+CTC、OCR-IC4D 和 OCR-IC4B 的效果差距不大，都接近与 99%，字符变长和定长并没有很大的影响最终的效果，而 OCR-IC8B 算法的效果相对更差，全字符准确率只有 86.4%，相比于其他算法准确率降低了 13%左右，这是因为字符长度从 4 增长到 8，增加了一倍的长度，因此对最终效果有所影响。

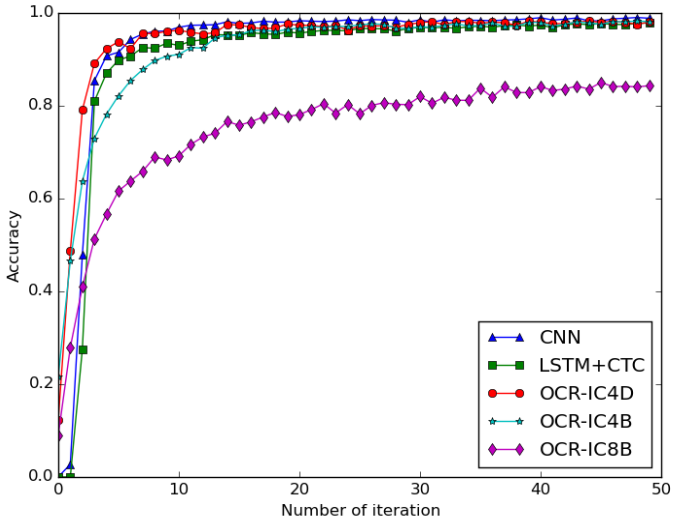


图 4-5 多个算法的迭代收敛图

表 4-1 是实验结果表，表中包含了 5 个算法在 2 个评价指标下的效果，为了让结果更加准确，使用了 10 万张验证码图片进行测试。从表中可以看到，当字符定长时，效果最好的是 CNN，本章提出的 OCR-IC 算法相比之下效果次之，但是优点是可以支持字符变长输入，并且即使字符变长时，效果也没有差异，三者效果最差的是 LSTM+CTC，该算法直接使用原始图像逐列像素作为输入，并没有使用卷积神经网络提取图像特征，可见卷积神经网络对于图像特征的提取是非常有效的。

表 4-1 实验结果

模型	字符长度	是否定长	准确率（单个字符）	准确率（全字符）
CNN	4	是	0.997	0.989
LSTM+CTC	4	是	0.991	0.970
OCR-IC4D	4	是	0.996	0.982
OCR-IC4B	4	否	0.996	0.985
OCR-IC8B	8	否	0.98	0.864

表 4-2 是 OCR-IC4D 算法在 10 万张验证码图片下单字符预测的混淆矩阵，从该矩阵中可以发现，对角线元素值非常大，说明预测结果绝大多数为正确结果，非对角线元素为预测错误的结果，从预测错误的结果中可以发现，预测错误最多的是第 7 行第 1 列，说明算法经常将数字字符“7”预测为“1”，这也符合预期，因为数字 7 和 1 的形状非常相似。

表 4-2 验证码混淆矩阵

		预测									
		0	1	2	3	4	5	6	7	8	9
真实	0	40016	43	46	6	22	4	25	41	18	26
	1	10	39698	8	4	22	12	6	145	4	5
	2	36	21	39488	11	41	0	7	151	11	25
	3	5	21	11	39997	5	50	6	33	58	28
	4	3	20	7	5	39898	9	9	30	4	14
	5	12	29	0	35	14	40079	134	13	29	5
	6	13	12	1	0	19	74	39804	11	34	3
	7	29	293	139	19	29	4	11	39496	4	17
	8	14	14	2	25	22	24	19	6	39879	14
	9	13	14	13	12	19	7	3	19	13	39341

4.3.4 实验结果可视化

本小节将对实验结果进行可视化，和第三章一样，先对图像特征提取部分进行可视化。本章图像特征提取所使用的卷积神经网络包含三个卷积层，第一个卷积层的卷积核大小为 5*5，如图 4-6 所示，左侧为原始验证码图像，包含 4 个数字字符“7957”，中间为第一个卷积层，右侧为提取的验证码图像特征。

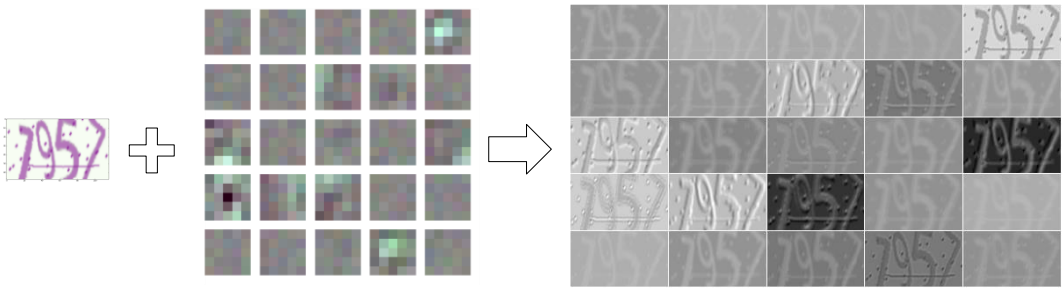
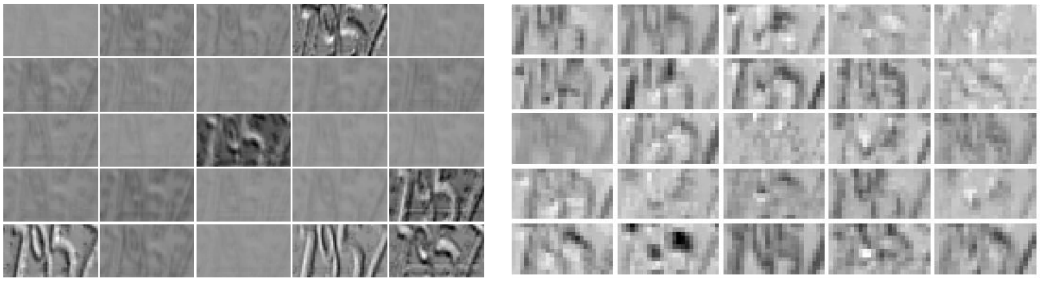


图 4-6 验证码图像特征提取



a）Conv2 层特征　　b） Conv3 层特征

图 4-7 验证码图像不同层卷积特征

图 4-7 中的子图（a）是第二个卷积层提取的图像特征，从该图中可以发现，有些特征图中的干扰线经过两次卷积后已经变得很不明显了。子图（b）是第三个卷积层提取的图像特征，相比于前两层，这一层的图像特征显得更加杂乱，可能是因为图像被放大所致，不过大致还能分辨出数字的轮廓。

图 4-8 是验证码图像标题生成样例，包含 16 个验证码图像，每张图像包含 1~8 个数字字符，图像下方是 OCR-IC 算法预测的结果。从图 4-8 中可以看到，大多数预测都是正确的，不过还是有少数的图片被预测错误，例如第一行第二列图片中的第三个数字 5 被预测成 6，第二行第三列的最后一个数字 3 被预测成 9。

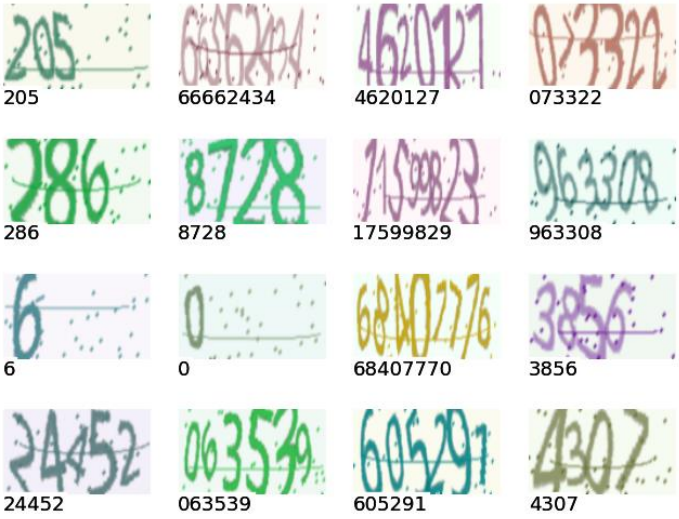


图 4-8 验证码图像标题生成样例

4.4 本章小结

本章主要介绍了如何以图像标题生成的角度，完成验证码识别的任务。本章首先介绍了数据来源和数据预处理，本章所有的验证码数据均使用 python 的 captcha 库自动生成，对生成的数据进行简单的预处理后，交给后续的步骤执行。第二节介绍了系统的整体设计思路，主要从三个方面进行阐述，算法的基本思想、系统的实现细节和模型的训练方法。由于验证码图片和真实场景图片略有区别，因此在算法设计中进行了针对性的修改，不仅修改了编码器中的卷积神经网络，还简化了解码器的整体结构。在进行代码实现时，将整个系统划分成多个模块，每个模块负责不同的功能，使代码结构更加清晰，接口调用更加灵活。对于模型的训练，本章使用了目前众多算法中性能较优的 Adam 算法，使算法可以更加快速的收敛。第三节介绍了对于验证码标题生成系统的实验，实验包含 3 个对比算法，2 种评价准则，最终实验结果表明，本章提出的验证码标题生成系统在多个方面有优越性，并且对于验证码的识别效果很不错。

结　论

人工智能是人类多年以来的梦想，如何让计算机能够看懂图像中的内容，并且以人类可以读懂的语言表达出来，也是目前学术界非常重要的一个研究方向。借助于近些年深度学习的火热，本文的研究课题图像标题生成算法也有了突破性的进展，不仅提取的图像特征更加有效，生成的自然语言文本也更加多样和灵活。本文通过对大量相关文献的调研，在已有的图像标题生成算法研究的基础上，提出了 past-feeding 和 past-attention 算法，两种算法分别基于不同的深度学习网络结构进行改进。此外，本文还将图像标题生成算法应用在验证码识别的问题中，相对于传统的解决方案，更加的通用和灵活，为验证码识别问题提出了新的解决方案。综上所述，本文的主要研究工作包含了以下几个方面：

（1）总结了传统的图像标题生成算法和基于深度学习的图像标题生成算法的研究现状，并对目前存在的问题进行总结和分析，同时阐述了传统验证码识别系统存在的弊端和以图像标题生成的角度来解决验证码识别问题的优势。

（2）在已有的 NIC 算法的基础上，本文针对性的提出了 past-feeding 算法那，通过引入 past-feeding 向量，将历史已经输出过的信息融合起来，辅助 LSTM 预测当前时刻的输出。实验结果表明，在预测过程中加入 past-feeding 向量的信息，可以一定程度上改善模型预测的正确性，在多个数据集以及评价指标上，past-feeding 算法均有不同程度的提高。

（3）在已有的 ATT-NIC 算法的基础上，本文针对性的提出了 past-attention 算法，通过将不同时刻的注意力向量建立联系，让下一时刻的注意力向量生成可以使用到历史的注意力信息，此外还将解码器中的语言信息部分和图像信息部分进行分离，使模型的分工更加清晰合理。实验结果表明，past-attention 算法相对于 ATT-NIC 算法有一定程度的提升。

（4）本文创新性的使用图像标题生成算法来解决验证码识别问题，并且针对验证码图片的特殊性，本文进行了一定程度的模型修改，使之更加契合验证码识别问题。最终实验结果表明，这种新的解决方案不仅在准确率上不输于其他的对比算法，并且能够很好的适应验证码长度不确定的情况，不需要人工干涉，完全端到端的学习。

本文提出的 past-feeding、past-attention 算法以及验证码标题生成系统存在需要进一步研究的问题：

（1）在 past-feeding 算法中，past-feeding 向量是将历史的输入信息加和得到，这种方式并不能区分历史输入信息中哪些是重要的，如果能够在 past-feeding 向量的计算上加入 attention 机制，相信能进一步提升模型效果。

（2）在 past-attention 算法中，使用软注意力算法作为基础，如何在硬注意力算法中加入历史注意力信息，也是值得研究的内容。

（3）目前本文的验证码标题生成系统的输入数据是由代码自动生成的，对真实场景中更加复杂、更多噪声、形式多样的验证码的研究工作仍然有待开展。

参考文献

[1] Hinton G E, Osindero S, Teh Y W. A Fast Learning Algorithm for Deep Belief Nets[J]. Neural Computation, 1960, 18(7):1527-54.

[2] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. Computer Science, 2015.

[3] Dalal N, Triggs B. Histograms of Oriented Gradients for Human Detection[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2005:886-893.

[4] Ojala T, Pietikäinen M, Mäenpää T. Gray Scale and Rotation Invariant Texture Classification with Local Binary Patterns[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2002, 24(7):971-987.

[5] Papageorgiou C P, Oren M, Poggio T. A General Framework for Object Detection[C]// International Conference on Computer Vision. IEEE Computer Society, 1998:573-5.

[6] Mori Y, Takahashi H. Image-to-Word Transformation Based on Dividing and Vector Quantizing Images With Words[C]// International Workshop on Multimedia Intelligent Storage & Retrieval Management. 1999:405-409.

[7] Duygulu P, Freitas N D, Barnard K, et al. Object Recognition as Machine Translation[J]. Eccv, 2002.

[8] Gupta A, Davis L S. Beyond Nouns: Exploiting Prepositions and Comparative Adjectives for Learning Visual Classifiers[C]// European Conference on Computer Vision. Springer-Verlag, 2008:16-29.

[9] Li L J, Fei-Fei L. What, Where and Who? Classifying Events by Scene and Object Recognition[C]//2007 IEEE 11th International Conference on Computer Vision. IEEE, 2007: 1-8.

[10] Yao B, Fei-Fei L. Modeling Mutual Context of Object and Human Pose in Human-object Interaction Activities[C]//Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on. IEEE, 2010: 17-24.

[11] Farhadi A, Hejrati M, Sadeghi M A, et al. Every Picture Tells a Story: Generating Sentences from Images[C]// European Conference on Computer Vision. Springer-Verlag, 2010:15-29.

[12] Kulkarni G, Premraj V, Ordonez V, et al. Babytalk: Understanding and Generating Simple Image Descriptions.[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2011:1601-1608.

[13] Mitchell M, Han X, Dodge J, et al. Midge: Generating Image Descriptions from Computer Vision Detections[C]// Conference of the European Chapter of the Association for Computational Linguistics. 2012:747-756.
[14] Srivastava N, Salakhutdinov R. Multimodal Learning with Deep Boltzmann Machines[J]. Journal of Machine Learning Research, 2014, 15(8):1967 - 2006.
[15] Kiros R, Salakhutdinov R, Zemel R S. Multimodal Neural Language Models[C]//ICML. 2014: 595-603.
[16] Mao J, Xu W, Yang Y, et al. Explain Images with Multimodal Recurrent Neural Networks[J]. Computer Science, 2014.
[17] Vinyals O, Toshev A, Bengio S, et al. Show and Tell: A Neural Image Caption Generator[J]. Computer Science, 2015:3156-3164.
[18] Xu K, Ba J, Kiros R, et al. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention[J]. Computer Science, 2015:2048-2057.
[19] Lecun Y, Bottou L, Bengio Y, et al. Gradient-based Learning Applied to Document Recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.
[20] Bouvrie J. Notes on Convolutional Neural Networks[J]. Neural Nets, 2006.
[21] Zeiler M D, Fergus R. Visualizing and Understanding Convolutional Networks[M]// Computer Vision – ECCV 2014. Springer International Publishing, 2013:818-833.
[22] Erhan D, Bengio Y, Courville A, et al. Visualizing Higher-Layer Features of a Deep Network[J]. 2009.
[23] Simonyan K, Vedaldi A, Zisserman A. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps[J]. Computer Science, 2014.
[24] Boureau Y L, Bach F, Lecun Y, et al. Learning Mid-Level Features for Recognition[C]// IEEE Conference on Computer Vision & Pattern Recognition. 2010:2559-2566.
[25] Krizhevsky A, Sutskever I, Hinton G E. Imagenet Classification with Deep Convolutional Neural Networks[C]//Advances in neural information processing systems. 2012: 1097-1105.
[26] Szegedy C, Liu W, Jia Y, et al. Going Deeper with Convolutions[J]. 2014:1-9.
[27] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
[28] Graves A, Mohamed A, Hinton G. Speech Recognition with Deep Recurrent Neural Networks[C]//IEEE international conference on acoustics, speech and signal processing. IEEE, 2013: 6645-6649.

[29] Weng C, Yu D, Watanabe S, et al. Recurrent Deep Neural Networks for Robust Speech Recognition[C]// IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE, 2014:5532-5536.

[30] Liu S, Yang N, Li M, et al. A Recursive Recurrent Neural Network for Statistical Machine Translation[C]// Meeting of the Association for Computational Linguistics. 2014:1491-1500.

[31] Auli M, Galley M, Quirk C, et al. Joint Language and Translation Modeling with Recurrent Neural Networks[J]. American Journal of Psychoanalysis, 2013:212-3.

[32] Elman J L. Finding Structure in Time[J]. Cognitive Science, 1990:179–211.

[33] Hochreiter S, Schmidhuber J. Long Short-Term Memory[J]. Neural Computation, 1997:1735-1780.

[34] Srivastava R K, Greff K, Schmidhuber J. Highway Networks[J]. Computer Science, 2015.

[35] Yao K, Cohn T, Vylomova K, et al. Depth-gated LSTM[C]//Presented at Jelinek Summer Workshop on August. 2015: 1.

[36] Koutník J, Greff K, Gomez F, et al. A Clockwork RNN[J]. Computer Science, 2014:1863-1871.

[37] Kalchbrenner N, Danihelka I, Graves A. Grid Long Short-Term Memory[J]. Computer Science, 2016.

[38] Ghosh S, Vinyals O, Strope B, et al. Contextual LSTM (CLSTM) models for Large scale NLP tasks[J]. 2016.

[39] Graves A, Fern&#, Ndez S, et al. Multi-dimensional Recurrent Neural Networks[C]// International Conference on Artificial Neural Networks. Springer-Verlag, 2007:549-558.

[40] Tan Y H, Chan C S. phi-LSTM: A Phrase-based Hierarchical LSTM Model for Image Captioning[J]. 2016.

[41] Cho K, Merrienboer B V, Gulcehre C, et al. Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation[J]. Computer Science, 2014.

[42] Chung J, Gulcehre C, Cho K H, et al. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling[J]. Eprint Arxiv, 2014.

[43] Greff K, Srivastava R K, Koutnik J, et al. LSTM: A Search Space Odyssey.[J]. IEEE Transactions on Neural Networks & Learning Systems, 2016.

[44] Gers F A, Schmidhuber J. Recurrent Nets that Time and Count[C]//Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on. IEEE, 2000: 189-194.

[45] Rush A M, Chopra S, Weston J. A Neural Attention Model for Abstractive Sentence Summarization[J]. Computer Science, 2015.

[46] Luong M T, Pham H, Manning C D. Effective Approaches to Attention-based Neural Machine Translation[J]. Computer Science, 2015.

[47] Chorowski J K, Bahdanau D, Serdyuk D, et al. Attention-based Models for Speech Recognition[C]//Advances in Neural Information Processing Systems. 2015: 577-585.

[48] Rocktäschel T, Grefenstette E, Hermann K M, et al. Reasoning about Entailment with Neural Attention[J]. 2015.

[49] Mnih V, Heess N, Graves A, et al. Recurrent Models of Visual Attention[J]. Computer Science, 2014, 3:2204-2212.

[50] Hermann K M, Kocisky T, Grefenstette E, et al. Teaching Machines to Read and Comprehend[C]//Advances in Neural Information Processing Systems. 2015: 1693-1701.

[51] Papineni K. BLEU: A Method for Automatic Evaluation of Machine Translation[J]. Wireless Networks, 2002:307-318.

[52] Flick C. ROUGE: A Package for Automatic Evaluation of Summaries[C]// The Workshop on Text Summarization Branches Out. 2004:25-26.

[53] Vedantam R, Zitnick C L, Parikh D. CIDEr: Consensus-based Image Description Evaluation[J]. Computer Science, 2015:4566-4575.

[54] Nesterov Y. A Method for Unconstrained Convex Minimization Problem with The Rate of Convergence O (1/k2)[C]//Doklady an SSSR. 1983: 543-547.

[55] Duchi J, Hazan E, Singer Y. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization[J]. Journal of Machine Learning Research, 2011:257-269.

[56] Zeiler M D. ADADELTA: An Adaptive Learning Rate Method[J]. Computer Science, 2012.

[57] Kingma D, Ba J. Adam: A Method for Stochastic Optimization[J]. Computer Science, 2014.

哈尔滨工业大学学位论文原创性声明和使用权限

学位论文原创性声明

本人郑重声明：此处所提交的学位论文《基于深度学习的图像标题生成算法与应用》，是本人在导师指导下，在哈尔滨工业大学攻读学位期间独立进行研究工作所取得的成果，且学位论文中除已标注引用文献的部分外不包含他人完成或已发表的研究成果。对本学位论文的研究工作做出重要贡献的个人和集体，均已在文中以明确方式注明。

作者签名：朱丹翔　　日期：2017年1月4日

学位论文使用权限

学位论文是研究生在哈尔滨工业大学攻读学位期间完成的成果，知识产权归属哈尔滨工业大学。学位论文的使用权限如下：

（1）学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文，并向国家图书馆报送学位论文；（2）学校可以将学位论文部分或全部内容编入有关数据库进行检索和提供相应阅览服务；（3）研究生毕业后发表与此学位论文研究成果相关的学术论文和其他成果时，应征得导师同意，且第一署名单位为哈尔滨工业大学。

保密论文在保密期内遵守有关保密规定，解密后适用于此使用权限规定。

本人知悉学位论文的使用权限，并将遵守有关规定。

作者签名：朱丹翔　　日期：2017年1月4日

导师签名：　　日期：2017年1月4日

致　谢

时光匆匆，如白驹过隙。在哈尔滨工大大学的两年半的硕士研究生学习生涯即将画上一个句号，在此向 ICES 辛勤工作的老师以及在我困难时给予帮助的同学们表示最衷心的感谢。

首先，感谢我的导师叶允明教授。本论文是在叶老师的亲切关怀和耐心指导下完成的，叶老师对于论文的选题、研究内容的确定、实验系统的搭建和论文的撰写与修改都给予了巨大的帮助，从这一系列过程中，我也深刻体会到叶老师一丝不苟、严谨治学的学术精神。此外，在生活上，叶老师给予学生无微不至的关怀，尽可能满足学生生活需求，让我们能够全身心的投入到学术当中，对此，再次对叶老师表示感谢。

感谢李丰师兄，在学习上给予我许多建议和帮助，对于课题中的许多难点，李丰师兄都能对我进行详细的解答。

感谢 ICES 的所有同学，我们一起学习，一起工作，一起为自己的未来拼搏奋斗，衷心的祝各位同学前程似锦。

最后，感谢我的父母，在我迷茫的时候给予我支持，在我开心的时候与我分享快乐，在精神与物质上对我都无条件支持。