U D C _____ 10486

武漢大學

硕 士 学 位 论 文

基于编解码框架的图像语义描述研究

研 究 生 姓 名: 陈新鹏

学 号: 2015202130029

指导教师姓名、职称: 姚剑 教授

专 业 名 称: 摄影测量与遥感

研 究 方 向: 图像语义描述

二〇一八年五月

Research on Image Semantic Caption Generation Based on Encoder-Decoder Framework

Candidate: Chen Xinpeng

Student Number: 2015202130029

Supervisor: Prof. Yao Jian

Major: Photogrammetry and Remote Sensing

Speciality: Image Semantic Captioning

School of Remote Sensing and Information Engineering

Wuhan University

May, 2018

论 文 原 创 性 声 明

本人郑重声明: 所呈交的学位论文, 是本人在导师指导下, 独立进行研究工作所取得的研究成果. 除文中已经标明引用的内容外, 本论文不包含任何其他个人或集体已经发表或撰写过的研究成果. 对本文的研究做出贡献的个人和集体, 均已在文中以明确方式标明. 本声明的法律结果由本人承担.

学位论文作者 (签名):

年　　月　　日

摘　　要

近年来，图像语义描述作为人工智能领域一项基本的研究任务，受到越来越多的关注。它作为桥梁连接了计算机视觉中的图像处理技术和自然语言处理中的序列语句生成技术。图像语义描述在实际生活中有着很多的应用，例如，它可以帮助视觉障碍者理解图像，也可以通过挖掘图像的语义内容来提高图像检索的质量。

图像语义描述任务也取得了很大的进展，尤其是基于编解码网络框架的模型在这个任务上取得了优异的性能表现。在本文中，我们提出了一种叫做自动重构网络（Auto-Reconstructor Network，ARNet）的网络结构，该网络嵌入于编解码网络模型之中，并且能够端到端的为图像生成描述语句。在我们的自动重构网络中，使用当前时刻循环神经网络产生的隐状态去重构前一刻时刻的隐状态，以此起到在不同时刻的隐状态之间进行信息迁移变换的作用。因此，通过自动重构网络，可以鼓励当前时刻的隐状态去从前一个时刻的隐状态中吸收更多有用的信息，并且能够挖掘相邻两个隐状态之间更深的语义关系，从而对循环神经网络中隐状态中信息的动态变换起到正则化的效果。

我们通过一系列的实验说明自动重构网络能够提升现有编解码网络模型的图像语义描述性能。同时，我们定性并定量的研究了解码网络在生成描述语句时训练阶段与测试推断阶段的差异性问题，发现我们的自动重构网络能够显著缓解这种差异性。此外，我们还将自动重构网络应用于置换顺序的序列化 MNIST 手写数字识别任务，同样显示了我们的模型能够对循环神经网络起到很好的正则化效果，尤其是在长依赖关系的建模上。

关键词: 图像语义描述, 编解码框架，卷积神经网络, 循环神经网络

ABSTRACT

Recently, image semantic caption generation has received increasing attention as a fundamental research problem in artificial intelligence. This tecnique works as a bridge which connects the image processing technique in computer vision and sequence generation in natural language processing. Generating descriptions of images automatically is very useful in practice, for example, it can help visually impaired people understand image contents and improve image retrieval quality by discovering salient contents.

Much advance has been made in image captioning, and an encoder-decoder framework has achieved outstanding performance for this task. In this paper, we propose a novel architecture, namely Auto-Reconstructor Network (ARNet), which, coupling with the conventional encoder-decoder framework, works in an end-to-end fashion to generate captions. ARNet aims at reconstructing the previous hidden state with the present one in recurrent neural networks (RNNs), besides behaving as the information transition operator. Therefore, ARNet encourages the current hidden state to embed more information from the previous one and expolits the deeper relationships between them, which can help regularize the transition dynamics of recurrent neural networks.

Extensive experimental results show that our proposed ARNet boosts the performance over the existing encoder-decoder models on image semantic captioning task. Additionally, we evaluate the discrepancy between training and inference processes for caption generation quantitatively and demonstrate that our ARNet remarkably reduces the discrepancy obviously. Furthermore, the performance on permuted sequential MNIST demonstrates that ARNet can effectively regularize RNN, especially on modeling long-term dependencies.

Key words: Image Semantic Captioning, Encoder-Decoder Framework, Convolutional Neural Networks, Recurrent Neural Networks

目　　录

1 绪论

1.1 选题背景与意义

近年来，深度学习（Deep Learning）技术在计算机视觉、自然语言处理以及语音识别领域取得了很大的成功。比如在大规模图像类别分类数据集 ImageNet [55] 上，最新的一些深度卷积神经网络（Convolution Neural Network, CNN），如 Inception-v4 [45]，NasNet [4]，其错误率分别为 4.8% 和 3.8%，已经低于人类 5.1% 的错误率了。又比如最近新的单词特征表示方法——Word2Vec [5] 技术及其延伸的 Paragraph2Vec [6] 技术，相对于传统的 One-hot 单词表示方法，不仅提升了单词的表示效果，同时也加快了模型的训练速度。原先可能需要一周才能完成训练的语言模型，现在只需要几个小时即可训练完成。正是因为这些技术的突破，深度学习也引领了这一轮人工智能（Artificial Intelligence, AI）的浪潮。我们国家也因此推出了《新一代人工智能发展规划》，标志了人工智能上升至国家发展战略的层面 [8]。

要实现人工智能需要我们攻克实现许多的技术难题，比如物体检测（Object Detection），场景语义分割（Scene Semantic Segmentation），以及图像语义描述（Image Semantic Captioning）等等。其中，图像语义描述任务是现阶段人工智能技术中一项基本的研究问题，它作为桥梁连接了计算机视觉（Computer Vision）中的图像处理技术以及自然语言处理（Natural Language Processing）中的语句生成技术，近年来吸引了众多的研究人员。图 1.1 展示了三幅图像语义描述的结果实例，其定义为给定一张自然场景下的图像，我们需要通过模型生成一句自然语句来描述图像的内容。要求所生成的描述语句不仅能理解并概括图像的内容，还要求通顺自洽，符合语言的语法。

图 1.1: 图像语义描述的结果实例

图像语义描述技术在实际生活中也有很多独特的应用场景，对于目前应用人工智能技术的产品落地部署，以此进入人们的生活有着指导性的意义。比如它可以帮助盲人或者视觉障碍者理解

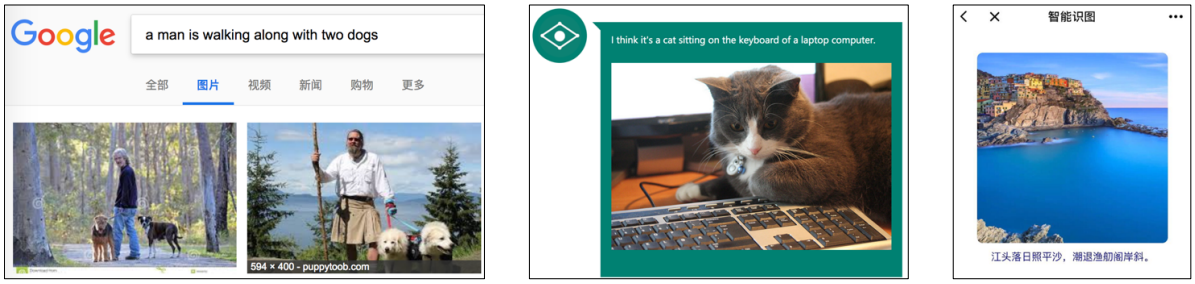

图 1.2: 图像语义描述技术在 *Google*、*Microsoft* 以及腾讯 *AI Lab* 中的应用实例

周边的事物、环境状况等。通过安装摄像头，这项技术可以实时的“告诉”盲人或视觉障碍者此刻他们身边有哪些事物，以及这些事物的状态。如图1.1的第二张图所示，通用物体检测模型或许可以检测出“cattle（牛）”，但图像语义描述模型所生成的句子不光能够“看到”牛，还能告诉人们这些牛正在吃草。通过这个例子可以看出，图像语义描述技术可以方便盲人及视觉障碍者的生活。此外，图像语义描述技术也可以用于辅助图像检索（Image Retrieve, IR）。目前基于文本的图像检索是最受人们接受的一种图像检索方式，即用户输入一句话，检索系统返回一系列相匹配的图像。起初，这些网络上图像的标注工作是由人工来完成的，虽然这样的人工标注能够取到较好的检索结果。然而，随着数据量的急速增长以及人工标注费用的快速增加，人工标注显然成为一种不现实的方式，而通过图像语义描述技术能够方便的匹配并检索到与文本语义相关的检索结果，同时降低了人工成本。又比如，在最近热门的对话机器人（Chat Bot）中，图像语义描述也是其中的核心技术之一，在与对话机器人“聊天”的过程中，用户不光可以通过输入文字进行聊天，还可以发送图片与对话机器人进行对话。目前，这项技术在许多科技巨头中已经有了许多具体的应用。图1.2中第一张图展示了图像语义描述技术在 Google 图像搜索引擎中应用，用户可以输入一句自然语句来搜索相应的图片；图1.2 中第二张图展示了图像语义描述技术在 Microsoft 的 CaptionBot 上应用的结果；图1.2中第三张图展示了腾讯 AI Lab 将图像语义描述技术应用于诗词的生成。

1.2 图像语义描述当前研究现状

1.2.1 自上而下的图像语义描述方法

目前，编码-解码框架（Encoder-Decoder Framework）是图像语义描述任务中最常采用的模型框架。这种框架由两部分组成，一是编码网络，二是解码网络。编码网络负责对输入的图像进行特征提取，获得输入图像的特征表示。然后由解码网络负责对提取得特征进行解码，生成图像的自然描述语句。也因为如此，这种编解码的模型也被称之为自上而下（Top-Down）的框架。

编解码网络框架最先被 Cho [9] 等人成功的应用在神经网络机器翻译（Neural Machine Translation，NMT）中。这种可以端到端优化学习的系统框架相比较于传统的基于词组的翻译方法相比，

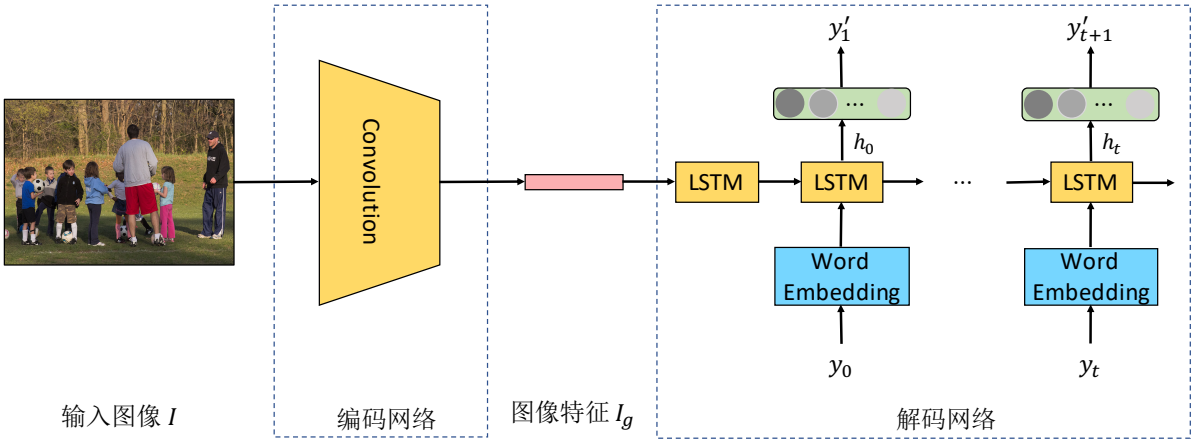

图 1.3: *Vinyals* [10] 等人提出的基于编解码框架的图像语义描述模型（*Neural Image Captioning, NIC*）示意图

显示了优异的性能，并被 Google 部署应用在自家的翻译系统上。受到此框架的启发，Vinyals [10] 等人首次将编解码网络成功应用于图像语义描述任务上并提出 Neural Image Captioning（NIC）模型，并在权威的 MSCOCO [18] 数据集上取得超越人类水平的描述结果。NIC 的模型结构如图1.3所示，编码网络由神经网络机器翻译模型中的循环神经网络（Recurrent Neural Network，RNN）[11–13] 替换为卷积神经网络，具体来说是 VGG [41] 网络模型，并且 VGG 模型的参数权重取自该网络在 ImageNet 上训练好的权重[1]；解码网络则是长短期记忆循环网络（Long Short-Term Memory, LSTM）[11]。具体来说，给定一张图像，记为 I。先由卷积神经网络对图像进行编码得到图像的全局特征，表示为 I_g。再将其进行输入到由 LSTM 单元组成的解码网络中，在 t 时刻，由上一时刻 LSTM 单元中的隐状态（Hidden State）h_{t-1} 以及输入的单词 y_t 得到当前时刻的隐状态 h_t，这里每一时刻的隐状态记录着之前时刻的语句特征信息。再通过一层全连接网络层（Fully Connected Layer，FC）与 Softmax 函数，将 h_t 映射到单词表长度的向量，该向量中的每个元素值即为每个单词的概率，然后我们选取最大概率的单词作为 t 时刻产生的单词 y'_{t+1}。依此类推，直至生成 `EOS` 时停止，此处的 `EOS` 代表了一句话的结束。自 NIC 这个工作之后，绝大多数图像语义描述的模型都是基于这种编码及解码框架。

但是 Mao [15] 等人注意到在上述的 NIC 结构中，只有在最初的 $t = 0$ 时刻用到了图像的特征信息（I_g），但实际上，在生成每一个单词时，均可以充分挖掘并利用图像的特征信息 I_g。基于此观察，他们提出一种多模态循环解码器（m-RNN）的模型。在该模型中，每一个 t 时刻，不仅使用 h_t 来生成单词 y'_{t+1}，而且融合了图像特征信息 I_g 以及当前时刻的输入单词 y_t 的词向量。

同年，Bahdanau [17] 等人首次提出了注意力机制（Attention Mechanism），并成功地将其用在神经网络机器翻译中，使机器翻译模型的性能取得很大的提升。Xu [16] 等人受之启发，首次在图像语义描述任务中引入并提出两种视觉注意力机制：软性注意力（Soft Attention）与硬性注意力（Hard

[1] http://www.robots.ox.ac.uk/~vgg/research/very_deep

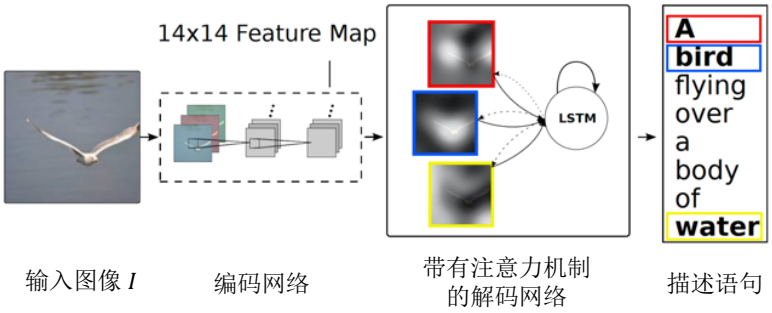

图 1.4: *Xu* [16] 等人提出的带有注意力机制的图像语义描述模型示意图

Attention）。如图1.4所示，在带有注意力机制的编码解码框架（Attentive Encoder-Decoder）中，从卷积神经网络中提取出的不再是图像的全局特征表示 I_g，而是图像的一系列局部特征表示。我们一般将卷积神经网络中最后一层卷积层的输出作为图像的局部特征，记为 $\mathbf{s} = \{s_1, s_2, \cdots, s_N\}$。接着在解码网络生成每一个单词时，带有注意力机制的网络模块能够动态的计算图像的每个局部特征（s_i）对预测单词的权重，最后生成新的图像特征输入到解码网络中。

基于 Xu 等人所提出的视觉注意力机制，其后产生了一系列的工作。Yang [50] 等人提出了性能更强的复习网络（ReviewNet），其模型框架如图1.5所示。在复习网络模型中，图像的一系列局部特征表示 $\mathbf{s}$ 以及全局特征表示 I_g 先被输入进复习网络模块中，复习网络经过多次对图像局部特征进行注意力操作，得到一系列所谓的想法向量（Thought Vectors），这些想法向量其实是多层次的图像特征表示。随后，同样带有注意力机制的解码网络再对这些想法向量的进行解码并生成描述语句。相比较于 Xu [16] 等人的解码网络模型，复习网络模型中的解码过程不再是直接对图像局部特征 $\mathbf{s}$ 进行注意力操作，而是在复习网络生成的一系列想法向量上进行注意力操作。

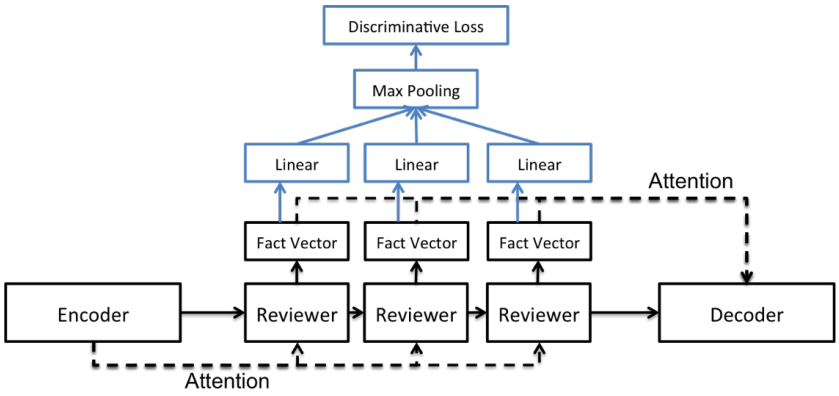

图 1.5: *Yang* [50] 等人提出的复习网络结构示意图

Lu [20] 等人注意到在描述语句中，有些单词诸如“of”之类的介词，“the”之类的定冠词是不需要借住图像信息的。基于此，他们在解码网络中，提出了一种所谓的空间注意力机制（Spatial Attention Mechanism）。具体思想是，通过网络的自动学习，让模型动态地决定什么时候需要引入更多的图像特征信息，什么时候需要引入较少的图像信息。具体的模型细节如图1.6所示，在传统视觉

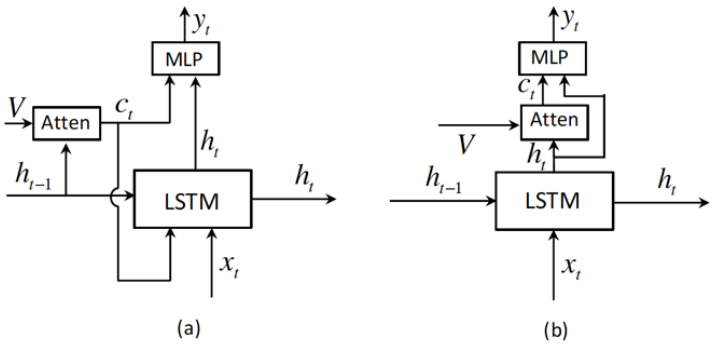

图 1.6: 左图（a）是传统的视觉注意力机制结构图，右图（b）是 $Lu^{[20]}$ 等人提出的空间注意力机制结构示意图

注意力机制的结构中，t 时刻时图像的每个局部特征的贡献权重由循环神经网络中上一个时刻的隐状态 h_{t-1} 来计算。但在 Lu [20] 等人的空间注意力机制模型中，t 时刻图像的每个局部特征的权重由当前时刻的隐状态 h_t 求得。

1.2.2 自下而上的图像语义描述方法

与 NIC [10] 这类基于编解码网络框架的方法不同的是，微软研究院的 Fang [24] 等人所提出的模型是一种自下而上（Bottom-Up）的方法。在他们的模型中，采用的是多示例学习（Multiple Instance Learning，MIL）[51,52] 方法，从所给的图像中检测提取可能存在的单词，并且可以把这些单词对应到图像的具体区域。所谓的多示例学习，其实是一种半监督（semi-supervised）或弱监督（weakly-supervised）的学习算法。使用多示例学习的数据由数据包（bag）组成，而每个数据包又由若干个示例（instance）组成。我们只有对数据包有正负类标记，对数据包里面的每个示例则没有正负类标记。当一个数据包被标记为正的时候，数据包中一定至少有一个示例是正样本。但对于被标记为负的数据包，它里面的所有示例一定都为负样本。我们的训练目标就是去学习一个分类器，该分类器能够对示例的正负进行判别 [23,25,26]。

在 Fang [24] 等人的模型中，先从图像语义描述数据集中得到单词的词汇表 $\mathcal{V}$（通常选取 1000 个出现频率最高的单词组成词汇表）。对于词汇表中的任意一个单词 w，如果人工标注的图像语义描述句子中含有该单词，则该图像就是正类的数据包。其中，图像中的每一个区域就是一个示例。然后采取迭代方法训练，先利用当前的模型权重，选出正类数据包中最有可能是正类的示例，以及负类数据包中的示例。然后利用选出的示例再进行训练，更新模型权重参数。依此类推，这样的迭代下去，便可以对图像的区域进行分类，也就可以从图片的区域中提取所需要的单词，如图1.7所示。若用 $\tilde{V}$ 表示从图像中提取到的可能出现的单词，然后我们通过语言模型，生成图像描述语句 $\{w_1, w_2, \cdots, w_n\}$。在生成第 i 个单词时，通过 $Pr(w_i|w_1, w_2, \cdots, w_{i-1}, \tilde{V}_i)$，其中 $\tilde{V}_i \in \tilde{V}$，表示生成第 i 个单词时，还没有使用过的单词。建模 Pr 的方法是从 $w_i, w_1, w_2, \cdots, w_{i-1}, \tilde{V}_i$ 这些单词中提取特征，如 w_i 是否属于 $\tilde{V}_i$，N-gram 关系等特征，记特征提取函数为 f_k。对每个要生成的单

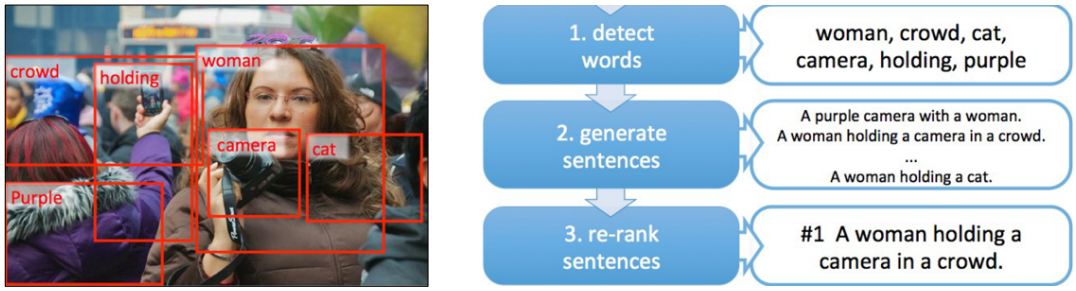

图 1.7: 左图是用多示例模型从图像中提取可能出现的单词结果示意图；右图为 *Fang* [24] 等人提出的模型流程图。整个模型分为三部分，第一步是从图像中检测出可能出现的单词，第二步是用语言模版生成一系列的描述语句，第三步是从上述的候选语句中选出最恰当的描述语句。

词，就可以得到一个分数：$\sum_k^K \lambda_k f_k\{w_i, w_1, w_2, \cdots, w_{i-1}, \tilde{V}_i\}$，最后针对一张图像，能够得到 M 句话。最后再使用 MERT [53] 算法进行得分排序，选取最好的一句话作为最终的生成结果。

1.2.3 自上而下与自下而上相结合的图像语义描述方法

尽管自上而下的编解码模型在最近几年中已经成为最主流的图像语义描述模型，但自下而上的传统模型仍有其用处。最近有许多工作成功的将这两种方法相结合 [19,27,54]，进一步提高了图像语义描述模型的性能。在 You [19] 等人的模型中，在用卷积神经网络提取图像特征的同时，他们提出了两种检测图像中视觉属性（visual attributes）信息的方法，一种是基于最近邻图像检索的无参数方法，另一种则是利用 Fang [24] 等人所提出的多示例学习方法。在检测出图像的视觉属性信息之后，他们将其注入注意力机制模块中，如图1.8所示。由于在注意力机制中混合了预先检测的视觉属性信息，所以这篇文章称这样的模型为语义注意力模型。

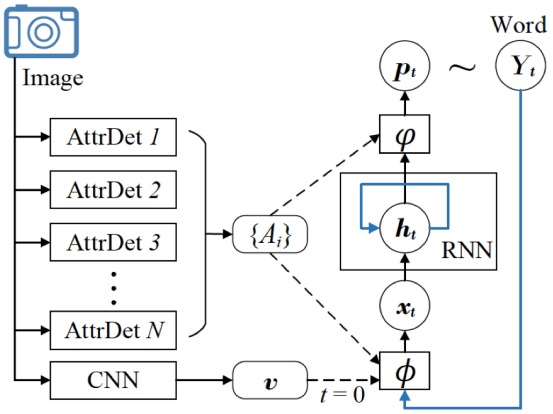

图 1.8: *You* [19] 等人提出的用视觉属性信息辅助视觉注意力的语义注意力机制模型

类似的，Yao [27] 等人所提出的模型也将图像的属性信息注入到编解码框架中。不同于 You [19] 等人的模型的是，You [19] 等人的方法是将检测得到的属性信息注入到注意力机制模块中，而 Yao [27] 则将属性信息用作一种补充信息融合进编解码框架中，同时提出了五种属性信息的融合方式，如

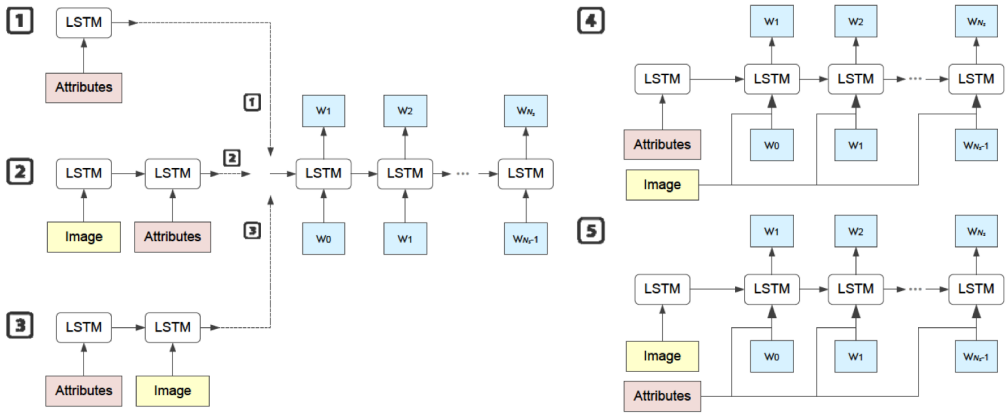

图 1.9: *Yao* [27] 等人提出的将图像属性信息融合注入到解码网络的模型

图1.9所示。

最近 Jiang [54] 等人提出的 LTG 模型跟 Yao [27] 等人的类似，都属于结合了图像视觉属性信息的模型。区别在于 Jiang 等人提出的 LTG 模型是基于 Yang [50] 所提出的 ReviewNet 模型，并将图像的属性信息通过一个引导网络（guiding network），注入到了复习网络的模块之中，LTG 模型的结构如图1.10所示。基于更强的复习网络，Jiang 等人模型的性能在 MSCOCO 数据集上取得进一步的提升。

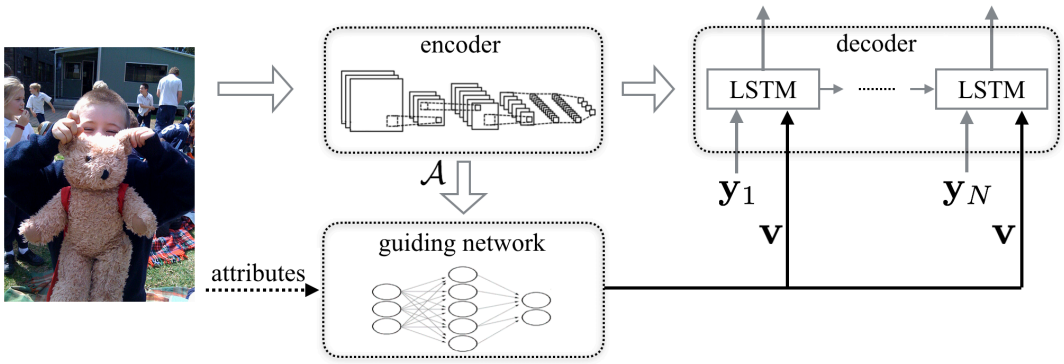

图 1.10: *Jiang* [54] 等人提出的 *LTG* 模型结构示意图

1.2.4 基于强化学习的图像语义描述方法

尽管在引入注意力机制后，再结合图像的属性信息，图像语义描述模型的性能已经有了很大的提升。但以上的工作所提出的图像语义描述模型，均有一个问题，即训练时的目标函数（object function）都是最大似然估计（Maximum Likelihood Estimation，MLE）损失函数，或交叉熵（Cross Entropy，CE）损失函数。也就是说，我们模型的训练目标是尽可能的去拟合真实数据（训练集）的分布。但我们在评测模型所生成的描述语句时，通常都是用的 BLEU [29]、Meteor [30]、CIDEr [31]、ROUGE [32]、SPICE [33] 这些评测标准，这种训练与评测标准的不一致会限制图像语义描述模型

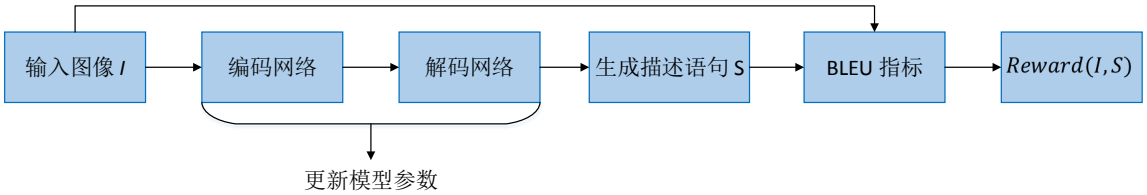

图 1.11: *Ranzato* [34] 等人提出的基于强化学习中的 *REINFORCE* 算法的模型，该模型直接以图像语义描述评测标准（如 *BLEU*）作为目标函数进行训练优化，缓解了训练目标与评测目标不一致的问题。

的性能。因此，Ranzato [34] 等人首次引入强化学习的方法，直接将 BLEU 指标的得分作为目标函数，于是模型的优化目标就是最大化 BLEU 评测指标的分数。同时，为了解决目标函数的不可微分导致不能直接用常用的梯度下降优化算法进行学习的问题，他们使用了 REINFORCE [35] 算法来计算近似梯度，由此便可以用梯度下降优化模型，模型框架如图。目前为止，在 MSCOCO[2]排行榜前列的，均是基于该算法或其变种形式，代表性的有 IBM 的 Rennie [37] 等人提出的自评判语言训练算法（Self-critical Sequence Training, SCST）等。

1.3 本文的主要工作及创新点

尽管现今基于编解码网络的图像语义描述模型在 MSCOCO 数据集上得到了很高的评测分数。但是，由循环神经网络所构成的解码网络仍然存在一个问题，那就是在模型训练时采用的是教学模式，即所谓的 teacher forcing 模式。在这种模式下，在训练阶段的 t 时刻，我们将人工标的描述语句中的单词 y_t 输入进循环神经网络，并由循环神经网络更新后的隐状态 h_t 来生成下一时刻的单词，记为 y'_{t+1}。然后我们计算 t 时刻人工标注的单词 y_{t+1} 与循环神经网络所生成的单词 y'_{t+1} 之间的交叉熵损失，从而对解码网络的进行监督训练。

但测试推断时，上述的图像语义描述模型则处于自由运行模式，即所谓的 free running 模式。在 $t=0$ 时刻，我们将起始单词 y'_0（实际上一般将 `BOS` 作为起始单词）输入到循环神经网络之中，将循环神经网络中的隐状态更新为 h_0，由 h_0 生成下一时刻的单词，记为 y'_1。但在 $t=1$ 时刻，我们则将 y'_1 输入到解码网络中，来生成下一时刻的单词，记为 y'_2。这个过程跟训练阶段不同，因为在测试阶段时，$t=1$ 时刻的正确的单词 y_1 我们是无法知道的。

但这样生成单词的过程中，当某一时刻生成的单词出错之后，下面的单词就都会出错，这便造成了解码网络中的训练阶段与测试阶段之间差异性问题（discrepancy problem），这种差异性还会引起曝光偏差（exposure bias）问题。为了应对这个问题，有两篇代表性的工作，一个是 Bengio [39] 等人提出的规定采样（Scheduled Sampling）算法，用该算法在训练过程中，就以一定的概率将正确的单词（如 y_t）用测试推断模式下得到的单词（如 y'_t）替代，即在训练时模拟测试推断的过程。

[2] `https://competitions.codalab.org/competitions/3221#results`

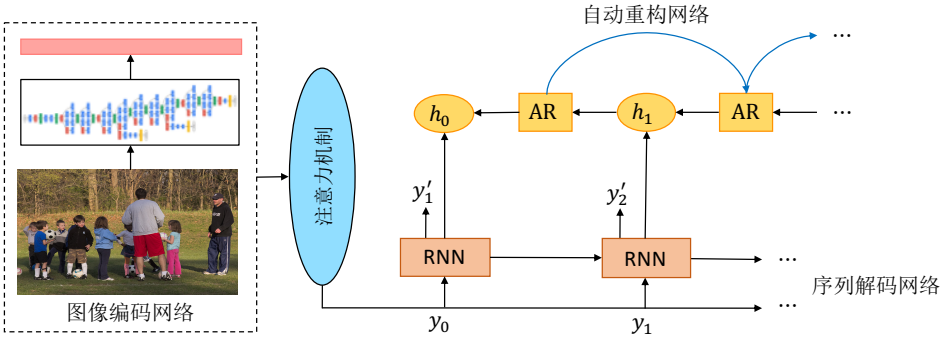

图 1.12: 本文所提出的自动重构网络结构（*Auto-Reconstructor Network, ARNet*）示意图

另一篇是 Krueger [40] 等人提出的 Zoneout 算法，该方法是在训练时，同样以一定的概率让解码网络选择是保留上一时刻的隐状态，还是像正常循环神经网络那样更新隐状态。这两类方法均可以理解为对循环神经网络进行正则，从而提高模型的性能。

在本文中，我们为了应对上述所讨论的问题，提出了一种叫做自动重构网络（Auto-Reconstructor Network, ARNet）的模型结构。如图1.12所示，自动重构网络嵌入于编解码网络框架之中，它作为桥梁连接了循环神经网络中相邻两个时刻的隐状态，如 $t-1$ 时刻的隐状态 h_{t-1} 与 t 时刻的隐状态 h_t。具体来说，我们使用 t 时刻的隐状态 h_t 去重构它前一个时刻的隐状态 h_{t-1}，重构的结果记为 h'_{t-1}。接着，我们用欧式距离来衡量重构后的隐状态 h'_{t-1} 与原始隐状态 h_{t-1} 之间的误差，以便用梯度下降优化算法进行训练与学习。

经过我们的自动重构网络，当前 t 时刻的隐状态 h_t 能够从前一个时刻的隐状态 h_{t-1} 中学习到更多的信息，增强了相邻隐状态之间的耦合性，同时能够对循环神经单元的信息迁移起到正则化作用。同时，为了说明本文所提出的自动重构网络能够显著缓解解码网络的训练阶段与测试阶段差异性问题。本文提出两种距离度量方式，用于定量的衡量解码循环神经网络中训练阶段的隐状态与测试阶段隐状态之间的差距。接着，本文所提出的自动重构网络能够显著提升图像语义描述的性能。在 MSCOCO 数据集上，与当前的非强化学习模型相比，使用本文所提出的模型性能够达到前沿的效果。最后，本文还在置换顺序的序列化 MNIST 数据集（Permuted Sequential MNIST）上进行实验，验证了我们的模型不仅对图像语义描述中的解码循环神经网络起作用，对一般的循环神经网络也能起到令人满意的正则化效果。

1.4 深度学习常用框架介绍与比较

自 2012 年由深度学习所引起的人工智能浪潮以来，不仅卷积神经网络、循环神经网络这些模型结构在快速迭代发展，深度学习框架也同样在迅猛发展，百花齐放。从最早的 Theano、Caffe、Torch 这些第一代深度学习框架，到 TensorFlow、Keras、PyTorch 这些最近发布的第二代深度学习框架。框架之多、更新迭代之快让深度学习研究者目不暇接。深度学习框架 Keras 的作者、谷歌

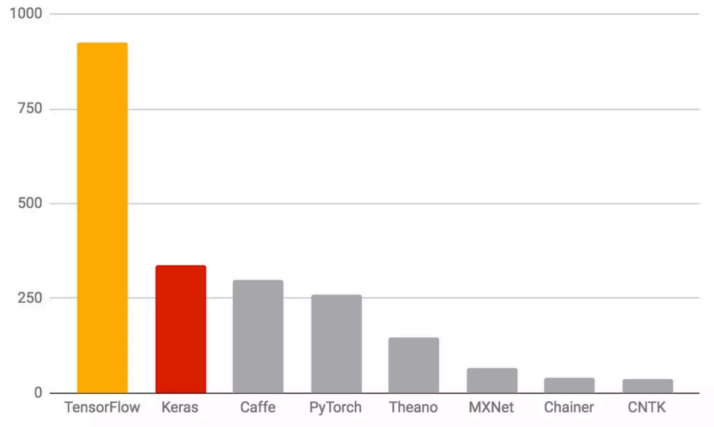

图 1.13: *François Chollet* 所统计的深度学习框架流行程度对比图

研究科学家 François Chollet 统计了 2017 年 12 月 7 日至 2018 年 3 月 7 日期间，ArXiv 网站上的论文中各自所使用的深度学习框架，统计结果如图1.13所示。从图中我们可以看出，TensorFlow 排名第一，Keras 排名第二，之后依次是 Caffe、PyTorch、Theano、MXNet、Chainer 以及 CNTK。本文在这一小节中将分别介绍常用的几款深度学习框架：Caffe、TensorFlow 以及 PyTorch。

1.4.1 Caffe

Caffe 全称为 Convolutional Architecture for Fast Feature Embedding，最初由伯克利大学的贾扬清于 2013 年开发并发布，现在由加州大学伯克利分校的计算机视觉中心（BLVC）继续维护。虽然 Caffe 属于第一代深度学习框架，但是它是第一个被工业界广泛接受并大规模用于落地部署的深度学习框架。尽管 Caffe 比较陈旧，但目前使用的企业、研究人员还是非常多的。对于工业界来说，Caffe 的优势在于能够方便的进行定制以便部署在不同环境的设备上，以此应用于不同的场景。而对研究人员来讲，Caffe 则显得较为落后、不够灵活，也没有第二代深度学习框架都有的自动求导机制。如果用户想定义自己提出的一个新的网络层或者对现有网络模型做一点改动，则还需要用户使用 C++ 和 CUDA 写出前向传播、反向传播的具体实现。同时，最初 Caffe 是专门设计用于计算机视觉方面的任务，对循环神经网络的支持不够友好，所以对于涉及到自然语言处理方面的任务显得力不从心。

1.4.2 TensorFlow

TensorFlow 属于第二代深度学习框架，支持自动求导机制。用户只需关注网络模型的定义，对于梯度的求导交给 TensorFlow 即可。同时它也是目前系统完整度最高的框架，支持绝大部分图像的卷积神经网络，同样支持自然语言处理、语音处理中常用的循环神经网络。最新版本的 TensorFlow 最大特点是同时支持先定义再执行（define and run）以及边定义边执行（define by run）。所谓的先定义再执行，又被叫做 lazy execution，它需要用户先将网络模型的执行图（execution

graph）先定义出来，然后再输入数据去运行。这样的机制在高并发高并行的场景下有天然的优势，能够保证运行效率，方便模型部署于各种各样的运行环境。正因为如此，现今工业界中深度学习产品的上线服务一般使用 TensorFlow 来部署。而所谓边定义边执行，又称作 eager execution，这样做的好处是灵活，方便调试。TensorFlow 受益于强大的社区支持以及 Google 的全力推广，很多深度学习最新的研究成果都是基于 TensorFlow 开发的，研究人员可以很快的使用到最新的模型。

1.4.3 PyTorch

PyTorch 亦是第二代深度学习框架，它是 2017 年初由 Facebook 人工智能实验室 FAIR 所推出的开源深度学习框架。PyTorch 最大的好处是其灵活性，能够方便用户在模型的执行的过程中对数据进行操作，数据在 CPU 与 GPU 之间的交互也非常便利。基于此，PyTorch 快速吸引了大批的深度学习研究人员。特别是在 NLP 领域，时常会需要构造动态模型图，如不等长的 LSTM 循环神经网络，这时候 PyTorch 就是当仁不让的选择。但在大规模分布式的应用场景下，PyTorch 目前还有些力不从心，还不如 TensorFlow 的稳定健壮。本文的图像编码部分使用 TensorFlow 来实现，因为可以借助 Google 提供的在 ImageNet 上预训练好的模型权重。而在对图像特征解码生成自然语句时，则采用 PyTorch 实现。

1.5 论文的内容和结构

本文共分为五个章节，每个章节的安排内容如下：

第一章：绪论。在第一节，文本先介绍了图像语义描述任务的选题背景与意义。在第二节，介绍了图像语义描述的研究现状，详细介绍了自上而下的基于编解码框架的模型方法；自下而上的基于图像视觉属性和语言模版的方法；引出了最大似然估计作为损失函数所引起的问题，由此介绍了基于强化学习的图像语义描述方法。在第三节，分析了编解码框架模型所引起的训练阶段与测试阶段差异性的问题，介绍了我们为解决这个问题所提出的自动重构网络。最后在第四节，本文介绍了三款当前广泛使用的深度学习框架。最后一节则介绍了本文各个章节的内容与安排。

第二章：基于编解码框架的图像语义描述。在第一节，本文先以经典的 LeNet-5 卷积网络为例介绍了卷积神经网络的基本概念与结构；接着介绍了本文所使用的 Inception-v4 图像编码网络；随后简要回顾了卷积神经网络的发展现状。在第二节，本文首先介绍了用于对单词进行特征压缩编码的词嵌入技术；接着介绍了循环神经网络的基本概念及结构，同时分析了一般循环神经网络所存在的问题，重点介绍了本文所使用的长短期记忆网络；再之后，本文介绍了普通的解码网络对编码特征利用不充分的问题，引入了视觉注意力机制，能够大幅度提升编解码模型的性能。最后，本文回顾了循环神经网络的发展现状。

第三章：自动重构网络。在第一节，本文分析了现今编解码网络中存在的曝光偏差问题及其发生的原因；随后第二节，引出了本文为解决这个问题所提出的自动重构网络，详细介绍了自动

重构网络模型的细节，同时将自动重构网络与 Scheduled Sampling 和 Zoneout 的进行了比较与分析。在最后一节，本文介绍了自动重构网络融入到编解码框架之后，整个模型的训练策略，以及我们采用这样训练策略的原因。

第四章：实验结果和分析。在第一节，本文介绍了用于图像语义描述任务的 MSCOCO 数据集、数据集的划分方案；介绍了四种常用的描述语句评价标准：BLEU、METEOR、ROUGE-L 以及 CIDEr。在第二节，本文介绍了本文中实验的参数配置。在第三节，本文对实验结果进行的讨论与分析。在第四节，本文通过定性的可视化展示，以及定量的分析，说明了自动重构网络对于缓解编解码网络中的曝光偏差问题有着较好的效果。接着通过实验讨论了不同大小的自动重构权重对模型性能的影响。最后本文将自动重构网络用于置换顺序的序列化 MNIST 手写数字分类任务，进一步展示了我们的方法对于一般任务中的循环神经网络均有着很好的正则化效果。

第五章：工作总结和展望。最后总结了本文的主要研究内容，并对图像语义描述任务的发展趋势与发展方向做了展望。

2 基于编解码框架的图像语义描述

如前文所述，对图像语义描述任务而言，本文所提出的自动重构网络是建立在编解码网络框架的基础上。而基于编解码网络的图像语义描述模型涉及到深度学习中两个基础又重要的技术，一个是用于对图像进行特征编码的卷积神经网络（Convolutional Neural Network, CNN）；另一个是对图像特征进行解码，生成自然描述语句的循环神经网络（Recurrent Neural Network，RNN）。

这一章共分为四节。本章的第一节介绍了用于对图像进行特征编码的卷积神经网络。先以经典的 LeNet-5 网络模型为代表，介绍了卷积神经网络的基本概念与结构；其次介绍了本文中所使用的 Inception-v4 卷积神经网络模型；接着介绍了当前卷积神经网络的发展与现状。本章的第二节介绍了用于生成自然描述语句的循环神经网络。在这一节中，先介绍了能够对单词高效编码的词嵌入网络（word embedding）；然后介绍了循环神经网络以及一般形式的循环神经网络所存在的缺陷；其次介绍了实际问题中最常使用的一种循环神经网络结构——长短期记忆网络（Long Short-Term Memory Network，LSTM）；然后详细分析了一般的解码网络对图像特征解码时，不能充分利用好图像的特征信息，于是引出了带有视觉注意力机制（visual attention mechanism）的解码网络，通过视觉注意力机制可以在生成描述语句时充分挖掘利用图像的特征信息；最后介绍了当前循环神经网络的发展与现状。

2.1 图像编码网络

2.1.1 卷积神经网络

图2.1是 LeCun 所提出的 LeNet-5 卷积神经网络，它是第一个成功应用于手写数字识别的神经网络，同时它有着现代卷积神经网络最基本的结构，本文以 LeNet-5 为例介绍卷积神经网络的基本构成。如图2.1所示，一般卷积神经网络由以下几种模块组成：

输入层（input layer）： 输入层是卷积神经网络输入数据的部分。在图像编码卷积神经网络中，它就是一张由图像像素组成的像素矩阵。在图2.1中，输入层输入的是一个维度为 $32 \times 32 \times 1$ 大小的三维矩阵。其中第一维度、第二维度分别代表了输入图像的宽度（width）与高度（height），其中第三维度代表了图像的深度（depth）或者通道数（channel）。在图2.1中，我们输入的是灰度图像，所以通道数为 1。但是在 RGB 模式的彩色图像下，图像的通道数则为 3。从输入层开始，卷积神经网络通过卷积核进行卷积变换操作，将上一层的三维矩阵变换为下一层的三维矩阵，这里的三维矩阵实际上是特征图（feature map）。

卷积层（convolution layer）： 卷积层是卷积神经网络最核心的结构。和传统的全连接层网

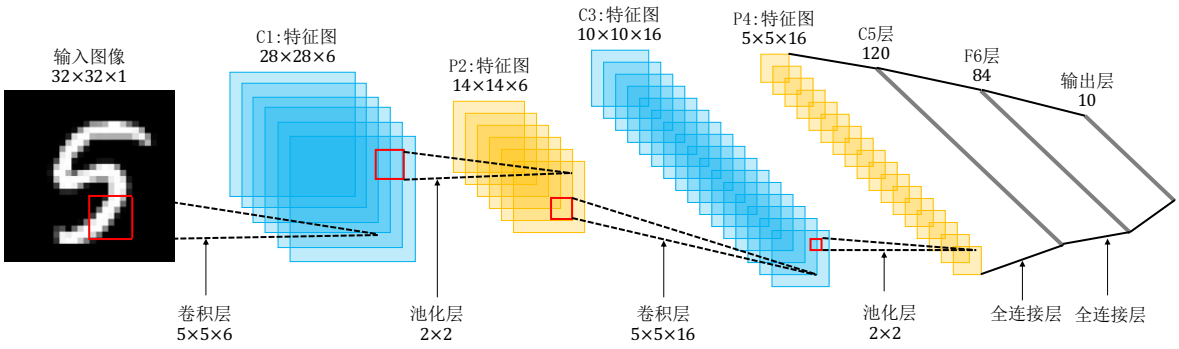

图 2.1: *LeCun* [1] 等人提出的 *LeNet-5* 模型结构示意图

络不同的是，在卷积层中每次卷积变换只是对上一层神经网络中的一小块进行操作。这种卷积操作是由名为过滤器（filter）或者卷积核（kernel）的结构来完成的。卷积核的长度与宽度是由人工设计并指定的，每个卷积核事实上还有深度这一维度，但与这一层特征图的深度相同。通常卷积核的长宽尺寸有 3×3 和 5×5，也有一些特殊的如 1×1，或者非对称卷积核如 1×3、3×1。如图2.1所示，LeNet-5 中的第一层卷积核的长宽为 5×5，其中图中所示的第三维度的值 6 代表卷积核的数量，同时也是输出的特征图的数量。更一般的，在某一卷积层中，对输入维度为 $W_1\times H_1\times D_1$ 的特征图进行卷积操作。其中，卷积核的长度和宽度均为 F，卷积操作的步长为 S，卷积核的数量为 K。为了避免尺寸的变化，我们可以对当前网络层矩阵的边界进行 0 填充（zero-padding），填充的边界长度记为 P，这样可以使得卷积层前向传播后输出矩阵的大小和当前网络层矩阵保持一致。经过这个卷积核操作之后，将输出的特征图维度记为 $W_2\times H_2\times D_2$，并有下列形式：

$$\begin{aligned} W_2 &= (W_1 - F + 2P)/S + 1, \\ H_2 &= (H_2 - F + 2P)/S + 1, \\ D_2 &= K. \end{aligned} \tag{2.1}$$

卷积运算也非常简单，图2.2展示了一个卷积操作的具体计算过程。在每一个位置，卷积操作的输出结果是卷积核与当前输入的局部图像的点积。相比于传统的全连接神经网络，卷积神经网络有如下特点：

- 局部连接（local connectivity）：又称稀疏连接（sparse connectivity）。传统的神经网络通过全连接层连接两个相邻的网络层，具体表现为当前网络层中的每一个神经元均与下一网络层的每一个神经元相连接，如图2.3左图所示。那么在图像处理任务中，当前网络层的每一个像素值均对下一层的结果产生影响。但事实上，对于一张图像而言，我们仅需局部的图像信息即可提取图像低层级的特征，比如图像的边缘、角点等等，而卷积神经网络的局部连接正好符合这个特点，如图2.3的右图所示。并且，一般认为人对外界的认知是从局部到全局的，而图像的空间联系也是局部的像素联系较为紧密，而距离较远的像素相关性则较弱。因而，每个神经元其实没有必要对全局图像进行感知，只需要对局部进行感知，然后在更高层将局部的

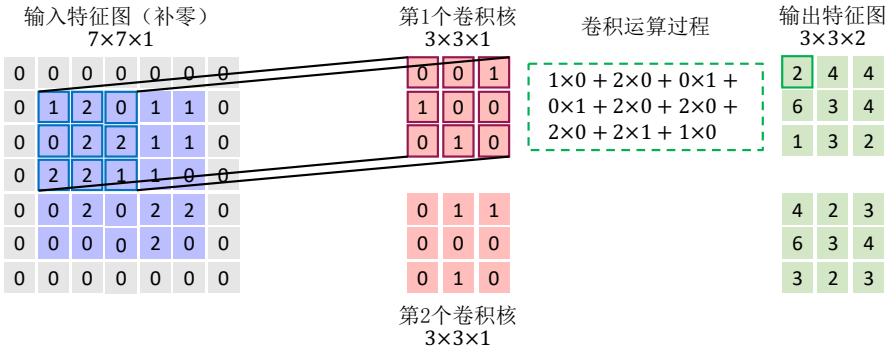

图 2.2: 卷积层前向传播计算过程样例图

信息综合起来就得到了全局的信息 [56]。

- 参数共享（parameter sharing）：使用全连接层处理图像的最大缺陷在于参数量过大。对于 MNIST 数据集而言，输入的图像尺寸是 $28 \times 28 \times 1$，如果使用全连接且设置下一层的神经元个数为 500，那么这一层的参数量将达到 $28 \times 28 \times 1 \times 500 + 500 = 392500$，而这仅仅是这一层的参数量。对于更加广泛的自然场景图像，图像输入的尺寸一般为 $224 \times 224 \times 3$，那么其参数量就为 $224 \times 224 \times 3 \times 500 + 500 = 75264500$。由此可见，如此参数量爆炸性增长的模型不仅仅无法训练或者训练速度极慢，还会容易导致过拟合的问题。但使用卷积网络之后，我们仅需要学习卷积核维度大小的权重即可，如同样使用 $224 \times 224 \times 3$ 的输入图像，卷积核大小为 $3 \times 3 \times 500$，那么使用卷积结构的网络模型参数量则为 $3 \times 3 \times 3 \times 500 = 13500$，即卷积的参数权重仅仅与卷积核的大小以及卷积操作输入输出的通道数有关，而与输入图像尺寸大小无关。所以卷积网络层的参数量相比于全连接层要少好几个数量级，因此，卷积网络模型的训练及存储上对于全连接网络模型有着极大的优势。并且从统计角度，所以对于这个图像上的所有位置，我们都能使用同样的特征。
- 层级结构（hierarchical structure）：卷积神经网络一般有多个卷积层组合而成，低层的卷积层用于提取低层次的特征，如边缘、角点等，而高层次的卷积层则用于提取更高层次、更全局的特征，如图像中物体的轮廓。低层的卷积层处理的是尺寸更大但语义层次较低级的图像特征，经过连续的卷积特征提取以及将采样过程之后，图像的尺寸逐渐缩小，到后面高层的卷积层，处理的就是尺寸更小但语义层次更高的图像特征。这样有如金字塔般的层级结构，使得对图像的理解过程是从局部到全局，从低层次的基础特征到更高层次的语义特征，这也更符合平常对事物的理解过程。

激活函数（activation function）：实际上，以上的卷积操作是一种线性变换，可以简化为 $y = \sum_{i=1}^{n} w_i x_i + b$，其中 b 为偏置项。但在实际生活中，简单的线性模型表达能力不够，无法拟合很复杂的数据。所以我们还需要加上一层非线性激活函数层，目的是引入非线性变换，增加网络模型的复杂度，使神经网络可以逼近任意复杂的函数。目前，常用的激活函数有如下几种：

- Sigmoid 函数：又称为 Logistic 激活函数，该函数的数学形式为 $f(x) = \frac{1}{1+e^{-x}}$，其函数曲

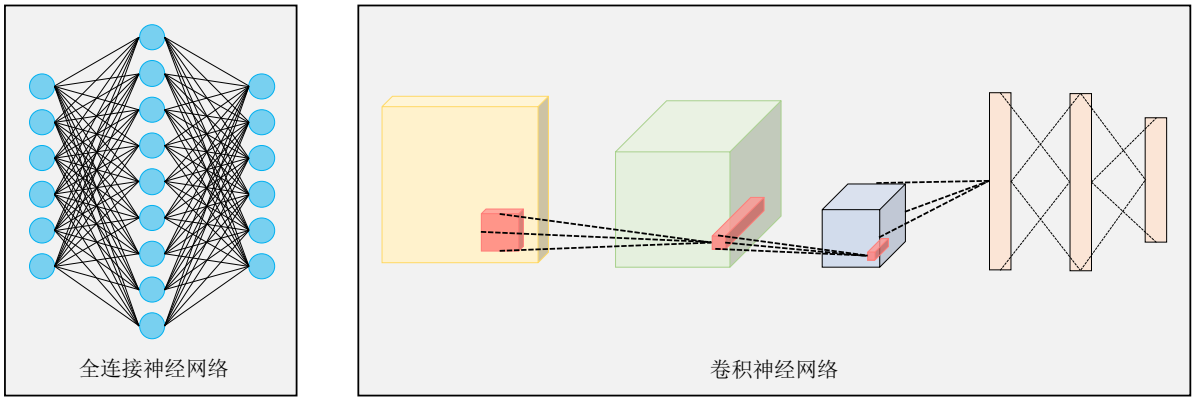

图 2.3: 全连接神经网络（左图）与卷积神经网络（右图）结构示意图

线如图2.4中所示。它可以将输入的实数值压缩至 0 到 1 的区间内，可以将大数值的负数压缩成 0，将大数值的正数压缩成 1。该函数最典型的应用场景是用于概率预测的问题中。但 Sigmoid 函数有三个问题，第一个问题，Sigmoid 函数会造成梯度消失。因为 Sigmoid 函数在趋近 0 和 1 的时候变化率会变得平缓，也就是说，Sigmoid 的梯度会趋近于 0，图2.4中 Sigmoid 函数的导数图直观的展示了这点。事实上，从图中可以看到，Sigmoid 函数梯度的最大值仅有 0.25。过小的梯度会导致相连神经元的权重更新得很慢，从而影响训练速度。第二个问题，Sigmoid 函数的计算成本相对较大，因为其中涉及到了以自然常数为底数的指数运算。第三个问题，Sigmoid 函数不是中心对称的函数。它所有的梯度均为正值，意味着所有的神经元权重更新时都沿着一个方向更新。即要么都沿着正方向更新，要么沿着负方向更新，这样的参数优化路径会导致模型的收敛速度变慢。

- Tanh 函数：又称为双曲正切激活函数（hyperbolic tangent activation function）。其实这个函数是 Sigmoid 函数的变形，该函数的数学形式为 $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$。从图2.4中可以观察到，Tanh 函数也是将输入值压缩在一个区间内，但是它是将输入压缩至 -1 到 1 的区间。与 Sigmoid 函数不同的是，Tanh 函数是中心对成函数，该函数输出值以零为中心。对于输入的负数，输出亦是负数，对于输入的正数，其映射值也同样为正数。在实践中，Tanh 函数的使用优先性一般高于 Sigmoid 函数。但同样的，Tanh 函数也会存在梯度消失问题，从图2.4中我们可以看到，Tanh 函数的最大梯度值为 1。同时，在输入值较大的地方，其梯度值很小并接近于 0，从而也会导致网络模型难以训练。
- ReLU 函数：又叫整流线性激活函数，该函数数学形式为 $f(x) = \max(0, x)$。当输入 $x < 0$ 时，输出都为 0，当 $x > 0$ 时，输出则为 x。实践证明，ReLU 函数能够使网络更快速地收敛。它不存在梯度信息的饱和问题，即它可以对抗梯度消失问题，至少在正区域（$x > 0$ 时）可以这样。同时，由于使用了简单的阈值化处理，ReLU 计算效率很高。但 ReLU 也存在几个问题，一是和 Sigmoid 激活函数类似，ReLU 函数的输出不以零为中心对称。二是如果 $x < 0$ 时，该函数梯度为 0，则无法传递梯度信息。为了解决这个问题，Maas 等人 [66] 提出了 Leaky

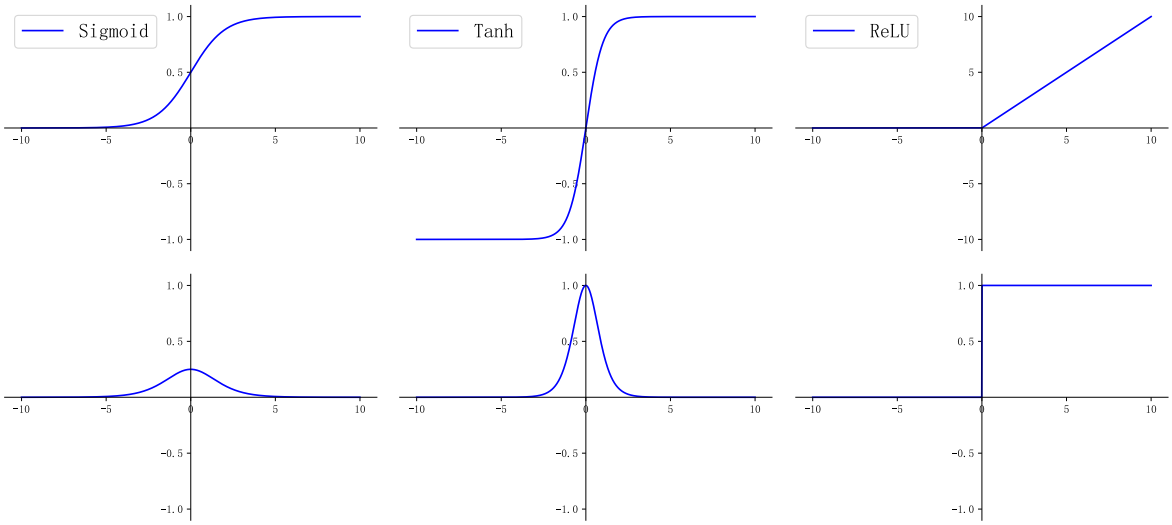

图 2.4: 三种主要的激活函数：*Sigmoid*、*Tanh* 以及 *ReLU*，以及它们对应的导数示意图

ReLU 函数，Leaky ReLU 函数是 ReLU 函数的改进版，其数学形式为 $f(x) = \max(0.1x, x)$，这样当 $x < 0$ 时，该函数也能够将梯度信息传递回去，缓解了 ReLU 函数所存在的问题。除了 Leaky ReLU，还有参数化的 ReLU 激活函数，该函数数学形式为 $f(x) = \max(\alpha x, x)$，该函数的特点是通过一个可以自动学习的超参数 α 来控制负数区域的输出。

池化层（pooling layer）： 从图2.1的 LeNet-5 网络模型中可以看到，在每一个卷积层之后，紧跟着还有一个名为池化层的网络层。池化层一般不会改变三维特征图的深度，但是可以有效的缩小特征图的尺寸。池化操作可以认为是将一张分辨率较高的图像转化为分辨率较低的图像。通过池化层，可以进一步的减少最后全连接层中神经元节点的个数，从而达到减少整个神经网络参数量的目的。这样既可以加速的网络的训练，也可以防止过拟合。池化层的前向传播过程也是通过一个类似于卷积核的过滤器来完成的，我们称之为池化函数（pooling function）。与卷积核中的加权求和操作相比，池化函数一般采用更加简单的取最大值操作或者求平均值操作。取最大值操作的池化层叫做最大池化层（max pooling），这是目前实践中使用最多的池化层结构。求平局值操作的池化层则叫做平均池化层（average pooling）。除了这两种，我们还有其他的池化函数，如基于 L^2 范数的池化层，还有基于据中心像素距离的加权平均池化层。

由卷积层、激活函数层、池化层和全连接层便可组成各种各样的卷积神经网络。正如前文所述，从最早的 AlexNet，到 VGG，GoogLeNet，再到 ResNet，Inception-v3，Inception-v4 等等，每一代网络都在进化，参数量变得更少，性能变得更强，速度也变得更快。本文中用于对图像进行特征编码的编码网络，使用的是在 ImageNet 数据集上预训练好的 Inception-v4 卷积神经网络。Inception-v4 卷积神经网络在 ImageNet 数据集上的 top-1 准确率是 80.2%，top-5 准确率是 95.2%，均属于非常领先的结果。因此，用 Inception-v4 模型所提取的图像特征能够很好的表示图像内容，为后面的解码网络提供更好的图像特征表示。

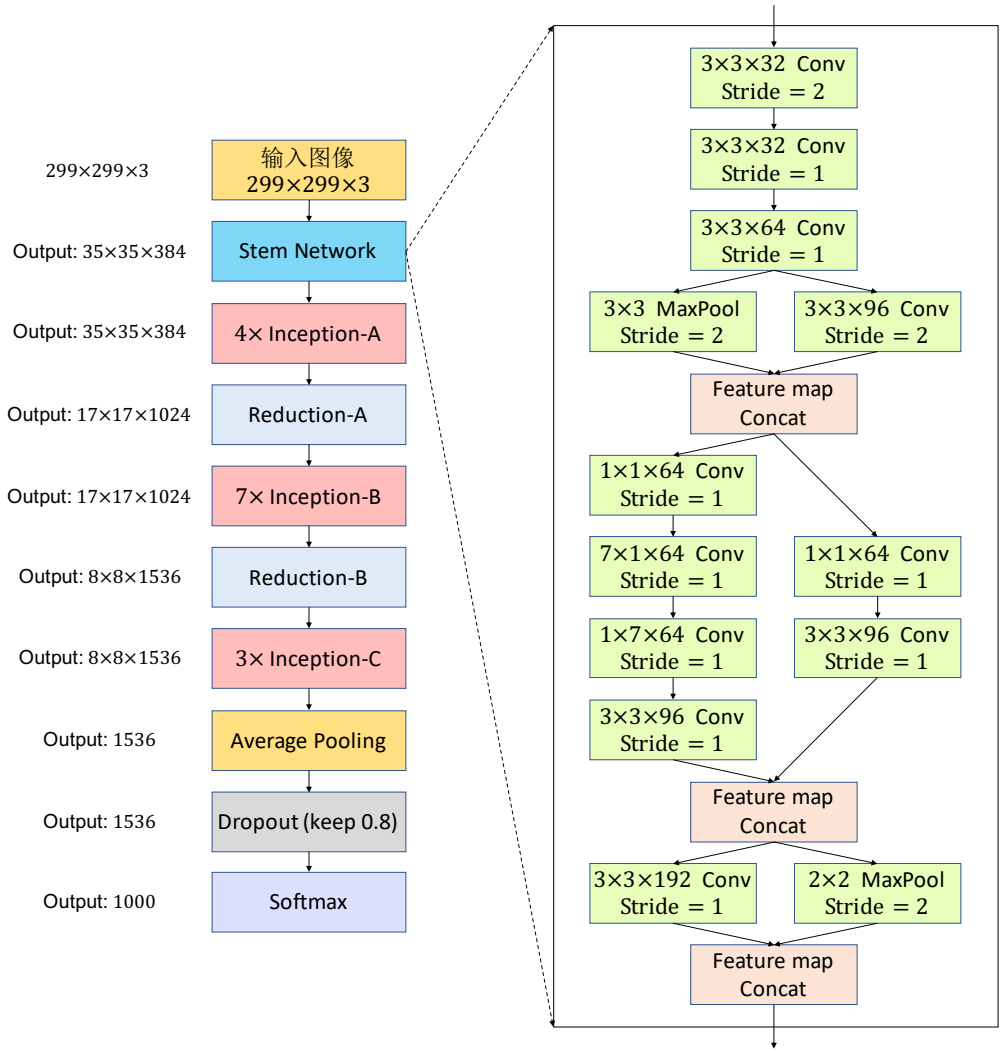

图 2.5: 左图是 *Inception-v4* 模型的整体框架，右图是主干网络（*Stem Network*）的详细结构图

2.1.2 Inception-v4 图像编码网络

Inception-v4 是 Inception-X [42–45] 系列网络中目前性能最强的一个模型。在这一小节中将介绍 Inception-X 系列网络模型所特有的 Inception 结构，以及 Inception-v4 卷积神经网络模型。Inception 结构是一种和传统的 LeNet-5、AlexNet、VGG 这些结构完全不同的网络结构。在 LeNet-5 这类传统网络结构中，不同的卷积层通过串联的方式连接在一起，而 Inception-X 系列网络中的 Inception 结构则是将不同的卷积层通过并联的方式连接在一起。

图2.5中的左图展示了 Inception-v4 卷积神经网络的整体结构。记输入图像为 I，先调整（resize）图像的尺寸至 $299\times299\times3$。接着，将图像 I 先输入进一个主干网络（Stem Network）进行处理，主干网络的细节展示在图2.5中的右图中。具体操作是将图像 I 经过三个连续的卷积层，然后经过两个并联子网络层，一个是最大值池化层，另一个仍然是一层卷积层，将它们的输出拼接起来。然后将拼接的结果输入进两个并联的卷积层，一边用较大尺寸的卷积核进行处理，另一边用小尺寸的卷积核进行处理，这种将不同尺寸的卷积核并联的结构就是所谓的 Inception 结构。Inception-v4

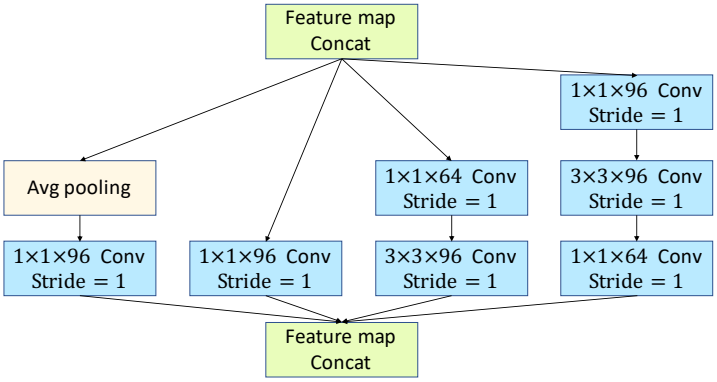

图 2.6: *Inception-A* 模块的详细结构图

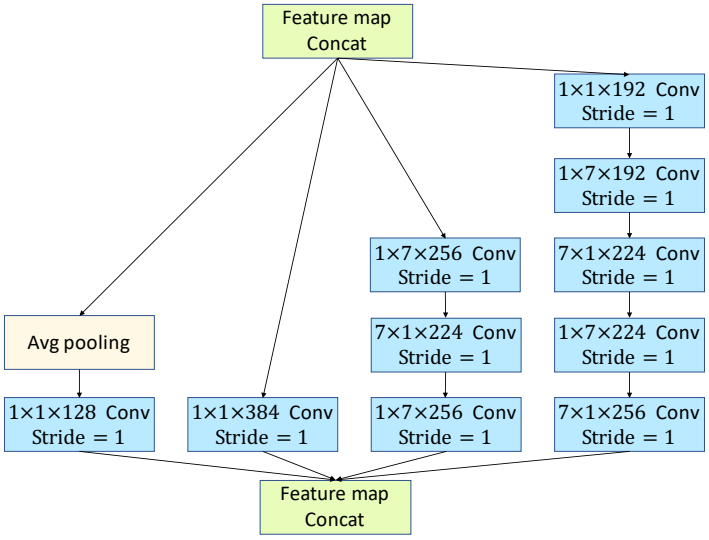

图 2.7: *Inception-B* 模块的详细结构图

相比较于前几个版本的 Inception 网络，最大的特点是将如 7×7 这样的卷积核分解成为 7×1 大小的卷积核以及 1×7 大小的卷积核，这样既减少了参数的数量，又加快了训练及测试推断的速度。在经过主干网络的处理之后，得到的特征图还会继续输入到多个连续的 Inception 模块进行进一步的处理，分别是四个连续的 Inception-A 模块，七个连续的 Inception-B 模块，以及三个连续的 Inception-C 模块。Inception-A、Inception-B 及 Inception-C 的详细结构分别如图2.6、图2.7及图2.8所示。注意到，最后一个 Inception-C 模块输出的是特征维度为 $8 \times 8 \times 1536$ 的特征图。我们在每个特征图上做平均值池化操作之后，便可得到一个维度为 1536 的特征向量，这个便可以作为输入图像 I 的全局特征表示，记为 g。同时，平均池化层之前的特征图维度为 $8 \times 8 \times 1536$，每一个位置可以作为图像的局部特征表示，记为 $\mathbf{s} = \{s_1, s_2, \ldots, s_N\}, s_n \in \mathbb{R}^{1536}$，其中 $N = 64$。这样的 64 个区域可以理解为将输入图像分割成 64 个局部区域，每一个区域 s_i 代表了原图像第 i 块区域的特征。在这里，提取图像每个区域特征的目的是为注意力机制模块做准备，后文会详细阐述。而对于一般的编解码网络模型，我们只需要得到图像的全局特征特征表示（g）即可。

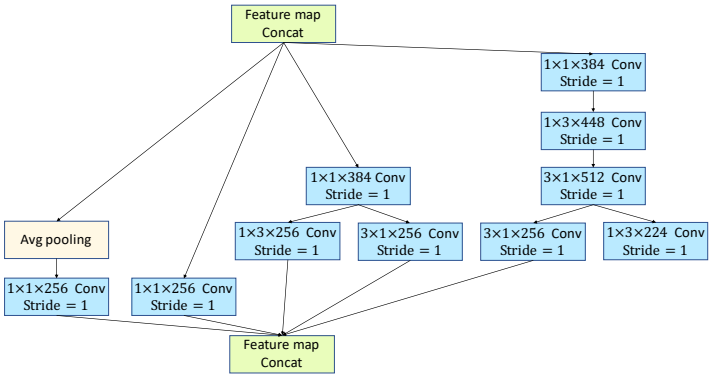

图 2.8: *Inception-C* 模块的详细结构图

2.1.3 卷积神经网络的发展与现状

卷积神经网络最早被成功应用于手写数字识别任务，1998 年 LeCun [1] 等人提出并实现了一个七层的卷积神经网络 LeNet-5 [2]。其模型结构如图2.1所示，LeNet-5 由两层卷积层（convolution layer）、两层降采样层或称池化层（pooling layer）、两层全连接层（fully connected layer），以及一层输出层（output layer）组成。在 MNIST 数据集上，LeNet-5 模型的错误率能够达到 0.8%。尽管 LeNet-5 在手写数字任务上取得了成功，也具有当今卷积网络基本的结构特征，但受制于当时计算机的计算能力以及支持向量机（SVM） [57] 的兴起，基于卷积神经网络的学习算法沉寂了很多年。

2012 年，来自 Hinton 组的 Krizhevsky [3] 等人在 ImageNet [55] 图像分类比赛上首次提出提出 AlexNet，该深度卷积神经网络模型相比较其他基于 SVM 的分类方法，以巨大优势取得冠军。AlexNet 显著的将 ImageNet LSVRC-2010 图像分类 top-1 和 top-5 测试集错误率从原先最好的 47.1% 和 28.2%，大幅度的降低到了 37.5% 和 17.0%。虽然 AlexNet 相比较于 LeNet-5，并不算是第一个使用卷积神经网络的工作，但 AlexNet 是第一个成功将深度卷积神经网络运用在 ImageNet 这种大型的数据集上的工作，自此深度学习开始受到人们的强烈关注，兴起了深度学习的热潮。同时，2012 年的 AlexNet 也是计算机视觉的研究热点从传统的视觉方法到深度学习算法的分水岭。同时在当年，Alexnet 是一个设计非常精巧的卷积神经网络模型，并且它提出了多种卷积网络训练技术，如使用整流线性激活函数（ReLU）解决梯度消失的问题，使用 dropout [61] 提高网络的泛化能力等等。

随后在 2014 年 ImageNet 图像分类任务的比赛上，来组 Google 与 Oxford 的研究人员分别提出了 GooLenet [42] 和 VGG [41]，并分别拿下了当年图像分类任务比赛的第一名与第二名。总的来说，VGG 网络是改进升级版的 AlexNet，它最大的特点是使用连续小尺度的卷积核（3×3）代替 AlexNet 中大尺度的卷积核（11×11）。这样不仅加深了卷积网络的深度，使得网络能够学习到图像更高层次的特征，并且减少了网络模型的参数量。GooLenet 因为其网络模型结构包含了 Inception 模块，因此也被成为 Inception-v1 网络模型。之后，Google 在 Inception-v1 网络模型

的基础上，增加了批规范化 [43]（batch normalization，BN）技术加快了网络的训练收敛速度，这个网络模型被我们称之为 Inception-v2。其后，Google 很快推出了 Inception-v3 [44] 网络模型，这个模型最大的特点是将 7×7 的卷积核分解为 1×7 和 7×1，这样既可以加速计算，又可以将一个卷积操作分解为两个，既加深了网络的深度，也增加了网络的非线性表示能力。而最新推出的 Inception-v4 [45] 网络模型则是当前众多图像编码网络中在 ImageNet 上表现性能最好的网络模型之一。本文也使用 Google 在 ImageNet 上预训练好的 Inception-v4 网络模型作为图像的编码网络。它的特点是融入了 He [46] 等人所提出的残差（residual）结构。残差结构是在网络层之间加入跳跃连接（skip connection），它缓解了梯度的消失问题，同时极大增加了网络的深度。基于残差结构的 ResNet 取得当年 ImageNet 图像分类比赛的第一名，其卷积网络的层数达到惊人的 152 层。除此之外，用于图像识别分类的网络模型还在不断发展，代表性的有 2017 年 CVPR 的最佳论文 DenseNet [48] 网络模型，以及 Google 最近提出的 NasNet [4] 网络模型等等。

2.2 序列解码网络

2.2.1 词嵌入网络

词嵌入（word embedding）[5,6] 网络也是图像语义描述网络模型的组成模块之一，它与下面所要介绍的循环神经网络一起构成了解码网络。词嵌入网络用于将用 One-hot 向量表示的单词映射到维度更低，但表示效率更高的特征向量。在自然语言处理中，传统的单词表示方式是 One-hot 表示法，这种单词表示法是将每个词表示为一个词汇表长度的一维向量。在这个向量中，绝大多数元素的值是 0，只有这个单词所在位置的值为 1。如现有“I love my country”这么一句话，词汇表就是句子中所有单词组成的集合，如此时的词汇表就为 [“I”, “love”, “my”, “country”]。那么，单词“I”在此词汇表中用 One-hot 可表示为 $[1,0,0,0]$，单词“love”可以表示为 $[0,1,0,0]$，单词“my”可以表示为 $[0,0,1,0]$，单词“country”可以表示为 $[0,0,0,1]$。但 One-hot 这种表示方式存在两个缺点，第一个缺点是维数灾难问题，当词汇表很大时，比如十万个单词的词汇表或者更多词汇的词汇表，那么每个单词用 One-hot 表示时，维度会非常的大，不利于存储与运算。第二个缺点是存在所谓的“词汇鸿沟”现象 [7]，即这样表示时的每个单词都是孤立的，仅仅从这两个向量中看不出两个词是否有关系。

在这里我们借助于词嵌入技术将用 One-hot 表示的单词进行降维操作，具体是通过单词嵌入矩阵（embedding matrix）来实现的。如图2.9所示，词汇表共计有 V 个单词，每个单词先用 One-hot 进行表示，则第 i 个单词 x_i 可以表示为 $[0,0,0,\cdots,1,\cdots,0,0,0] \in \mathbb{R}^{1\times V}$。再用词嵌入网络将其维度降低到 N 维，记嵌入矩阵为 $M \in \mathbb{R}^{V\times N}$。可以发现，单词 x_i 正好可以用词嵌入矩阵的第 i 行进行表示，同时将特征维度降低到了 N 维。嵌入矩阵的参数权重一般先用均匀分布随机初始化，连同整个编解码网络模型一起进行训练得到。

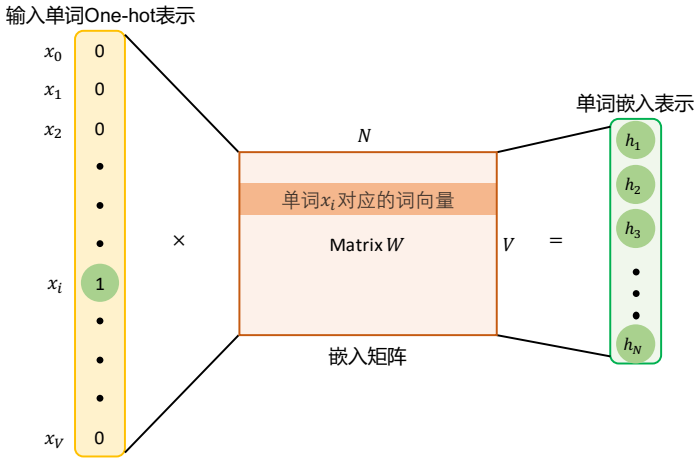

图 2.9: 词嵌入（*word embedding*）网络示意图

2.2.2 循环神经网络

生成如描述语句般的序列，最通用的做法是使用循环神经网络（Recurrent Neural Network，RNN）。相比于卷积神经网络这种前馈传播的网络结构，循环神经网络最大的特点在于存在时间维度的传播。如图2.10所示，它所展示的就是一个基本的循环神经网络。在 t 时刻时，循环神经网络会有一个输入，记为 x_t。然后结合循环神经网络前一个时刻内部的状态 h_{t-1}，会有一个输出 h_t。我们又称 h_t 为隐状态（hidden state），它既包含了当前输入的 x_t 的信息，也包含了前面时刻所记忆的信息。之后，h_t 又会作为 $t+1$ 时刻循环神经单元的输入，与 $t+1$ 时刻的输入 x_{t+1} 共同生成 $t+1$ 时刻的输出 h_{t+1}，依此类推。所以，循环神经网络可以看作同一个神经网络结构在时间序列上被复制多次的结果，这个被复制多次的结构被称之为循环体。和卷积神经网络中每一层卷积层中卷积核的参数共享类似，在循环神经网络中，循环体网络结构中的参数在不同时刻也是共享的。

如图2.10所示，循环神经网络中的循环体是通过两层全连接层来实现的，对于 t 时刻的输入 x_t 以及上一个时刻循环体的隐状态 h_{t-1}，分别将这两个输入用全连接层进行线性变换，这两个全连接层的参数分别记为 W_{x2h} 和 W_{h2h}，求和之后通过 tanh 激活函数层，得到 t 时刻时循环体的隐状态 h_t。以上过程的数学形式如下：

$$h_t = tanh(W_{h2h}h_{t-1} + W_{x2h}x_t) \tag{2.2}$$

但这样的循环神经网络在实际运用中效果并不好，很长时间都没有办法应用到实际问题。因为这样的结构在用随机梯度下降算法优化时，存在梯度消失（gradient vanish）以及梯度爆炸（gradient explosion）的问题，即目标损失函数所求得的梯度信息无法传递到前面时刻的循环体上。导致循环神经网络难以记忆较长时刻的信息，亦会导致网络难以训练。因为如果将起始时刻 $t = 0$ 时循

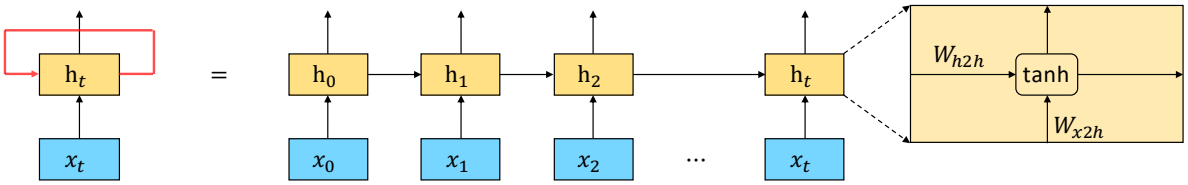

图 2.10: 一个简单的循环神经网络示意图，它是沿着时序方向展开的神经网络。其中，最右图是循环体内部的实现细节，通过两个简单的全连接层来处理循环体的两个输入 x_t 和 c_{t-1}，全连接层的参数分别是 W_{x2h} 以及 W_{h2h}，激活函数是 $tanh$ 函数。

环体的隐状态 h_0 传递到 t 时刻，则有：

$$\begin{aligned} h_t &= tanh(W_{h2h}h_{t-1} + W_{x2h}x_t), \\ &= tanh(W_{h2h}(tanh(W_{h2h}h_{t-2} + W_{x2h}x_{t-1})) + W_{x2h}x_{t-1}), \\ &= tanh^t(W_{h2h})^t h_0 + \cdots \end{aligned} \tag{2.3}$$

将 W_{h2h} 进行特征分解：

$$W_{h2h} = Q\Lambda Q^T, \tag{2.4}$$

其中，Q 是正交矩阵，那么循环体可以简化为：

$$h_t = tanh^t(Q\Lambda^t Q^T h_0) + \cdots . \tag{2.5}$$

从上式中可以发现，当特征值小于 1 时，经过 t 次相乘后，得到的结果会衰减到零；而如果特征值大于 1 时，经过 t 次连续相乘，其结果则会快速增长。无论哪种情况，都很难将 h_0 的信息传递到 t 时刻循环体的隐状态 h_t 上。

同时，上述形式的循环神经网络还会带来另一个问题，即长期依赖（long-term dependencies）问题。在有些简单的问题中，序列模型仅仅需要前几个时刻的信息来判断当前时刻的输出。比如预测“海洋的颜色是蓝色”这句话中最后一个单词“蓝色”时，序列模型仅仅需要看到“海洋”这个单词，即可预测出“蓝色”，并不需要更之前的信息来辅助预测。在这类应用场景中，我们所要预测的内容与相关的上下文信息间隔很短，普通的循环神经网络模型能够较容易地捕捉并保存临近时刻的上下文信息。但在实际更多的应用场景中，我们需要捕捉之前更长时刻的上下文信息来辅助预测当前的输出。比如当序列模型要预测这样一句更复杂的话“当站在月球上回望我们的地球，会发现地球是蓝色的”，如果我们要预测准确最后一个单词“蓝色”时，仅仅借助前一刻时刻的单词“地球”是无法预测准确的，因为可以预测为“地球是土黄色”，甚至可以说“地球是球形的”等等。这种情况下，我们需要借助更久之前的信息来预测当前的单词。普通的循环神经网络无法学习到距离较远的两个时刻信息之间的关系，使得循环神经网络的应用受到极大的限制。

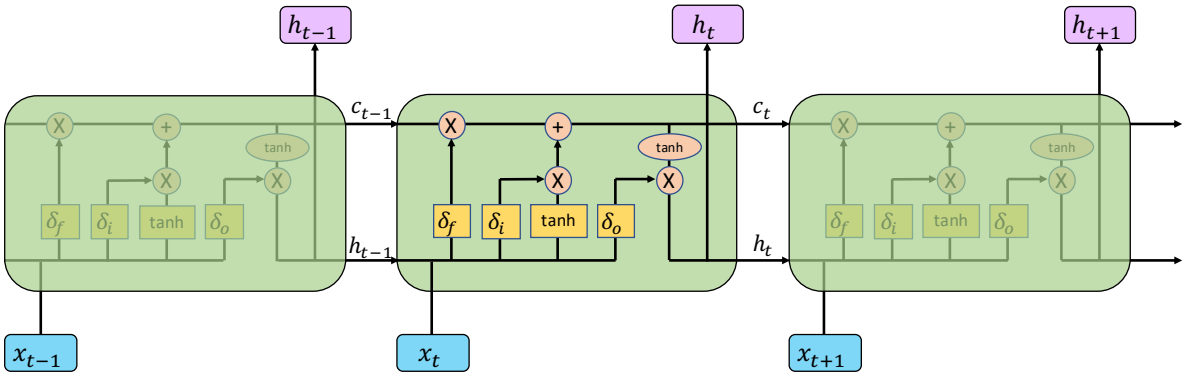

图 2.11: *LSTM* 循环体结构示意图

2.2.3 LSTM 序列解码网络

为了解决普通 RNN 所存在的问题，Hochreiter 和 Schmidhuber 于 1997 年提出一种全新的循环体结构——长短期记忆网络（Long Short-Term Memory Network, LSTM）。经过实践证明，在很多任务上，采用 LSTM 的循环神经网络的模型比一般的循环神经网络模型在绝大多数的任务上性能表现更好。如图2.11所示，与上述仅将 tanh 函数作为激活函数的循环体不同，LSTM 是一种有着三种特殊“门”结构的循环神经网络。

所谓的“门”结构，就是将输入的向量先通过全连接层进行线性变换，然后再通过 Sigmoid 激活函数层得到输出结果。由于 Sigmoid 函数输出的是 $(0,1)$ 之间的数值，它可以描述当前输入的向量有多少信息量是必要的。这种结构的设计如同一扇信息“门”，当 Sigmoid 函数的输出是 1 是，信息门会打开，让输入信息流入循环体中进行处理；当 Sigmoid 函数输出结果是 0 时，信息门则会关闭，阻止输入信息流入循环体中。如图2.11所示，LSTM 循环神经网络中有三种门结构，分别是：遗忘门（σ_f）、输入门（σ_i）以及输出门（σ_o）。

遗忘门 σ_f 的作用是让循环神经网络忘记之前记忆单元 c_{t-1} 中没有用的信息。但遗忘门是开起状态还是关闭状态，则由当前时刻的输入 x_t，以及前一个时刻循环神经网络的输出 h_{t-1} 共同决定。具体来说，先将 x_t、h_{t-1} 分别通过一层全连接层进行线性变换并求和，然后将求和结果通过 Sigmoid 激活函数得到结果。

在用遗忘门选择性的遗忘掉之前循环体记忆状态 c_{t-1} 的一部分信息之后，我们还需要用当前输入的信息 x_t 对记忆单元中的信息进行补充。具体的，我们将当前的输入同样类似的经过一个输入门 σ_i，进行输入信息的过滤，选取有用的信息，忽略无用的信息。输入门状态的决定过程与遗忘门类似，也是由 x_t 以及 h_{t-1} 经过一层全连接层及 Sigmoid 激活函数来实现，不过输入门的参数独立于遗忘门全连接层的参数。

通过遗忘门及输入门，LSTM 可以通过网络的自动学习，来决定哪些信息应该保留，哪些信息应该舍弃。在更新完当前时刻循环体内部的记忆状态 c_t 后，LSTM 还会产生当前时刻的输出 h_t，这个输出是通过输出门 σ_o 来控制的。同样的，输出门也是由当前的输入 x_t 以及上一时刻的输出

h_{t-1} 经过另一组权重独立的全连接层及 Sigmoid 函数来实现。

上述的过程即为 LSTM 的工作方式，具体的数学表示式如下：

$$\begin{aligned} f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_i), \\ i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_f), \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o), \\ c_t &= f_t \odot c_{t-1} + i_t \odot tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c), \\ h_t &= o_t \odot tanh(c_t) \end{aligned} \tag{2.6}$$

其中，f_t、i_t、o_t 分别为遗忘门、输入门以及输出门。c_t 为当前 t 时刻循环神经网络内部的记忆状态，h_t 为当前时刻循环神经网络的输出结果。σ 为 Sigmoid 激活函数层。W_{xf} 和 W_{hf} 为遗忘门的全连接层权重参数；W_{xi} 和 W_{hi} 为输入门全连接层的权重参数；W_{xo} 和 W_{ho} 为输出门全连接层的权重参数。$\odot$ 为逐元素相乘运算。b_i、b_f、b_o 及 b_c 为偏置项。

但事实上，上式可以进一步简化，可以写为如下的形式：

$$\begin{aligned} \begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} &= \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathbf{T} \begin{pmatrix} x_t \\ h_{t-1} \end{pmatrix}, \\ c_t &= f_t \odot c_{t-1} + i_t \odot g_t, \\ h_t &= o_t \odot \tanh(c_t), \end{aligned} \tag{2.7}$$

上式中，f_t、i_t、o_t、c_t、h_t 和 σ 分别为遗忘门、输入门、输出门、记忆状态、隐状态、Sigmoid 激活函数。矩阵 T 为线性变换矩阵，也即为上述过程中全连接层的参数权重。只不过这里的实现过程中，将上述的全连接层整合在了一起。因为对于 GPU 来说，将多个小矩阵相乘合并为大矩阵相乘要比串行计算多个矩阵乘法快很多。因此，本文中将多个门网络层合并到一个大的网络层来加速计算。

我们的图像语义描述模型中的解码网络正是基于 LSTM 循环神经网络来实现的。给定一张图像 $\mathbf{I}$，在经过 Inception-v4 图像编码网络之后，我们得到一系列图像局部特征表达 $\mathbf{s} = \{s_1, s_2, \ldots, s_N\}$ 以及图像的全局特征表达 g，我们的目的就是用 LSTM 循环神经网络对图像特征进行解码，生成一系列由单词组成的句子 $\mathcal{C} = \{y'_1, y'_2, \cdots, y'_T\}$。要求生成的自然描述语句不仅自然通畅，而且要能够准确抓住图像的语义内容。

本文先讨论如图1.3所示的最基础的编解码网络模型。在这个模型中，当我们得到图像的全局特征表达 $g \in \mathbb{R}^{1536}$ 后，先用一层全连接网络层对图像特征进行降维操作。然后将其输入到 LSTM 循环体单元中。这种情况下，可以认为是用全局特征 g 对 LSTM 进行初始化，此时的时刻记为 -1。然后从第 0 时刻开始，我们将当前时刻单词的嵌入词向量输入到 LSTM 单元中，得到每个时

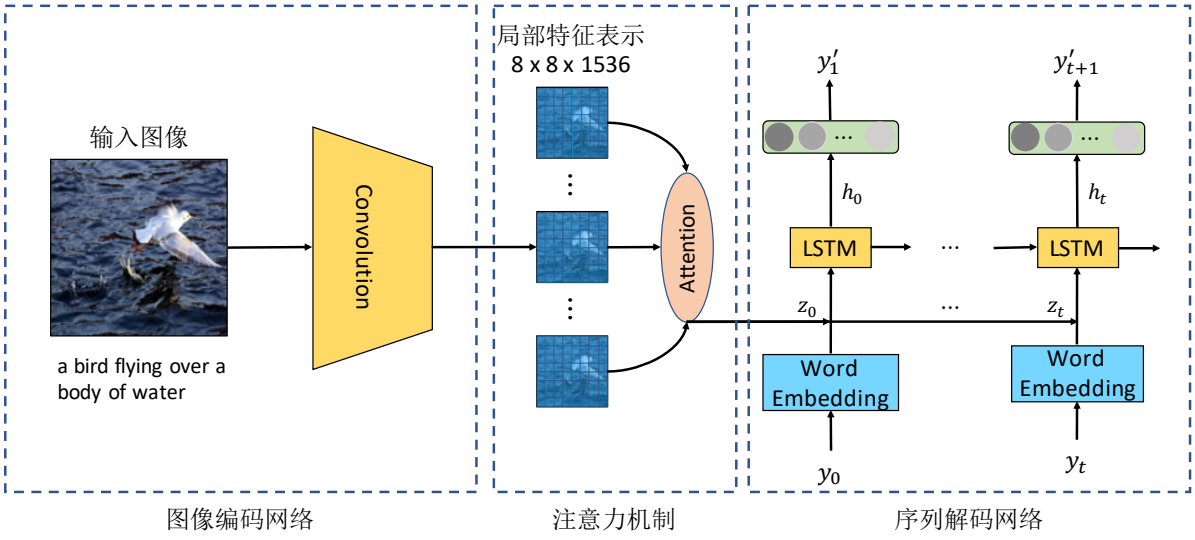

图 2.12: 带有注意力机制的图像语义描述模型框架

刻 LSTM 输出的隐状态 h_t。再将这个时候的隐状态 h_t 用一层全连接层映射到词汇表长度的向量，接着对这个向量通过 Softmax 层，得到每个单词出现的可能性（概率）。最后，我们用 Argmax 操作来选取概率最大的单词作为当前时刻预测的结果。

2.2.4 带有视觉注意力机制的序列解码网络

上述简单的编码解码描述模型中存在一个缺陷，即解码网络只有在最开始的时候用到了图像特征 g。但在后续生成每个单词的过程中，输入到 LSTM 中的只有单词的嵌入词向量，一个单词接着一个单词的生成，最终形成一句话。但正如前文所述，即便是 LSTM 这样的精心设计的循环神经网络，它的记忆能力还是有限的。随着时间轴的推进，越往后生成单词时，所能记住的图像特征 g 会越来越少。但如果将每个时刻的图像特征直接输入到 LSTM 循环体中也并不合理，因为当我们对一张图像进行描述，生成每个单词的过程中，图像中不同区域的特征对生成单词的贡献（权重）是不一样的。如图2.12 所示，当要预测单词“bird”时，图像中含有鸟的区域对模型预测时的影响应该较大，其他不相干的区域权重应该较小；当要预测单词“water”时，图像中含有水面的区域对预测的贡献应该最大，这时候非水面区域的图像特征对预测的权重应该较小。

在带有注意力机制的图像语义描述模型中，在解码网络的每一个时刻 t，我们会用前一刻时刻 LSTM 循环体输出的隐状态 h_{t-1}，以及图像局部特征 $\mathbf{s} = \{s_1, s_2, \cdots, s_N\}$ 中的每一个局部特征 s_i，通过一个全连接网络层来计算一个权重 α_i。这个权重的大小反应了图像的每个局部区域特征 s_i 在 t 时刻对预测单词的影响。最后输入到 LSTM 单元中的图像特征记为 z_t，实际上它是在 t 时刻每个区域的特征表示（s_i）与各自区域权重（α_i）的加权乘积之和。具体来说，z_t 计算过程的数

学形式如下：

$$z_t = f_{att}(\mathbf{s}, h_{t-1}) = \sum_{i=1}^{|\mathbf{s}|} \frac{\exp\left(\alpha(s_i, h_{t-1})\right)}{\sum_{j=1}^{|\mathbf{s}|} \exp\left(\alpha(s_j, h_{t-1})\right)} s_i, \tag{2.8}$$

其中，f_{att} 代表上述中的注意力机制函数。$\alpha(s_i, h_{t-1})$ 则是根据上一时刻的隐状态 h_{t-1} 决定图像第 i 块局部特征权重的函数。在本文中，$\alpha(s_i, h_{t-1})$ 通过多层感知机（multi-layer perceptron，MLP）来实现，其数学形式如下：

$$\alpha(s_i, h_T) = W_{s_i,h_T} \tanh(W_{s_i} s_i + W_{h_T} h_T), \tag{2.9}$$

上式中，W_{s_i,h_T}、W_{s_i} 以及 W_{h_T} 是感知机的参数权重。

于是，对于带注意力机制的图像语义描述模型来说，在 t 时刻时，LSTM 序列解码网络的有四个输入，分别是当前时刻输入的单词 y_t，注意力机制得到的加权过后的图像特征 z_t，上一个时刻 LSTM 单元的状态 c_{t-1} 和输出的隐状态 h_{t-1}。其数学形式如下：

$$\begin{aligned}
\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} &= \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathbf{T} \begin{pmatrix} y_t \\ h_{t-1} \\ z_t \end{pmatrix}, \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t, \\
h_t &= o_t \odot \tanh(c_t),
\end{aligned} \tag{2.10}$$

上式中，f_t、i_t、o_t、c_t、h_t 和 σ 分别为遗忘门、输入门、输出门、记忆状态、隐状态、Sigmoid 激活函数。矩阵 T 为线性变换矩阵。在每个时刻得到 LSTM 输出的隐状态 h_t 之后，将其线性映射到词汇表长度大小的向量，经过 Softmax 操作之后，即可得到词汇表中每个单词出现的概率，接着再取 Argmax 操作，得到 t 时刻生成的单词。以上的过程数学形式如下：

$$\begin{aligned}
p_{t+1} &= \text{softmax}(W_{y'} h_t), \\
y'_{t+1} &= \text{argmax}(p_{t+1}).
\end{aligned} \tag{2.11}$$

其中，$W_{y'}$ 为将 LSTM 输出的隐状态 h_t 映射到词汇表长度向量的全连接层参数。经过若干个时刻，当我们生成的单词为结束符号 `EOS` 时，停止生成句子。我们再将每个时刻生成的单词连接起来，即可得到图像描述的最终结果，$\mathcal{C} = \{y'_1, y'_2, \cdots, y'_T\}$。

2.2.5 循环神经网络的发展与现状

最早的循环网络是由 Hopfield [58] 于 1982 年提出，我们称之为离散 Hopfield 神经网络（DHNN，Discrete Hopfield Neural Network），因为这个网络的神经元的输出只能是 1 和-1，属于二值神经网络。但早期的循环神经网络在用反向传播算法时受到梯度消失或者梯度爆炸的困扰，难以训练以

及运用到实际应用中。于是，Jürgen Schmidhuber 等人在 1997 年提出了长短期记忆（Long Short Term Memory network, LSTM）神经网络。LSTM 相比较于简单的循环神经网络，最大的特点是循环神经单元内部设计了三个门单元结构以及一个记忆单元结构（memory cell）。三个门结构分别是遗忘门（forget gate）、输入门（input gate）以及输出门（output gate），遗忘门控制每个时刻循环神经单元中上一个时刻的信息有多少在当前时刻保留，输入门控制每个时刻输入的信息有多少要输入到循环神经网络的单元中，输出门控制当前循环神经网络单元的信息（即记忆单元）有多少需要输出。LSTM 有效的缓解了循环神经网络中的梯度消失的问题，但同样受制于当时的数据量、计算能力，训练方法，LSTM 同样沉寂了很多年。同年，Schuster [59] 等人提出了双向循环神经网络（Bidirectional recurrent neural networks），该网络由两层反向的循环神经网络组成，每一时刻的隐状态由正序传播的循环神经网络和倒序传播的循环神经网络组成，这样的结构既考虑了过去的信息，也考虑的未来的信息。

循环神经网络在序列模型的学习中超过传统方法的工作同样来自 Hinton 组。2013 年来自 Hinton 组的 Graves [60] 将循环神经网络用在语音识别任务上，其模型性能首次大幅度超过传统方法，奠定了循环神经网络在序列相关任务中的地位。在 2014 年，Bengio 组的 Chung [13] 提出 LSTM 的改进版：门控循环神经网络（Gated Recurrent Neural Networks，GRU）。GRU 相比较于 LSTM，它将遗忘门、输入门和遗忘门改进为两个门：更新门（update gate）以及重置门（reset gate）。同时，将 LSTM 中的记忆单元与隐状态统一合并成一个隐状态表示。尽管如此，在许多任务中，GRU 也能够达到 LSTM 的效果。并且 GRU 相比较于 LSTM，参数量小，计算速度更快，所以当我们要考虑到硬件成本、计算速度时，也会选择 GRU 作为序列建模工具。但将循环神经网络大规模部署到某个具体的应用上时（比如机器翻译），仍会受到计算速度的限制。Bradbury [14] 等人注意到，诸如 LSTM 这样的循环神经网络，每个时刻 LSTM 中三种门计算需要依赖上一时刻的输出隐状态 h_{t-1}，所以 LSTM 网络需要每个时刻依次计算，无法并行计算，这样导致了效率变慢。因此，他们提出了一种新的循环神经网络结构——Quasi-RNN。他们设计的循环体类似于 LSTM，但与 LSTM 不同的是，他们将三种门计算设计简化成只依赖当前的输入 x_t 的结构。这样，在 LSTM 中的每个时刻，各个门的计算可并行计算，这样便大幅度提高了循环神经网络的运算速度。

在应用方面，循环神经网络不光成功应用于语音识别领域，还在许多任务上相比较于传统方法取得大的突破，如机器翻译 [17]，代码语义标注 [50]，图像语义标注 [10,16] 等等。事实上，以 LSTM 和 GRU 为代表的循环神经网络已经成为时序相关任务中基础性的建模工具。

3 自动重构网络

尽管带有注意力机制的编解码网络模型及其各种各样的改进版本在图像语义描述任务上已经取得了很大的进展，但编解码网络模型仍然存在着一个缺陷，那就是网络模型的训练阶段与测试推断阶段之间所存在的差异性问题，也就是所谓的曝光偏差（exposure bias）问题。在本章的第一节中，介绍了所谓的曝光偏差问题，并且详细分析了产生这个曝光偏差问题的原因。在第二节中，介绍了我们为了解决这个问题所提出的一种全新的网络模型——自动重构网络（Auto-Reconstructor Network, ARNet），并详细介绍了自动重构网络的模型结构与实现方式，同时讨论了自动重构网络与规定采样（Scheduled Sampling）、Zoneout 这两个循环神经网络正则化算法之间的联系与区别。最后的第三节，介绍了我们提出的自动重构网络的训练策略，同时详细分析并解释了采用这样训练策略的原因。

3.1 编解码模型所存在的问题

在所有基于深度学习方法的模型中，模型都有两个阶段，一个是训练阶段，另一个就是测试推断阶段，基于编解码网络的图像语义描述模型也不例外。而曝光偏差问题就是由训练阶段模型的行为模式与测试推断阶段模型的行为模式不一致所导致的。

当编解码网络模型处于训练阶段时，在 t 时刻，我们需要将由人工标注句子中对应的单词 y_t 输入到 LSTM 循环体单元中。将 LSTM 网络输出的 h_t 经过一层全连接网络层线性映射后，经过一层 Softmax 层，可以得到当前时刻词汇表中每个单词出现的可能性（概率），我们用一个与词汇表长度等长的向量来表示，记为 p_{t+1}。因为在训练阶段，我们能够知道 t 时刻正确的单词应为 y_{t+1}（实际上我们为了计算损失函数，事先将其用 One-hot 向量表示），所以我们便可通过交叉熵（Cross Entropy, CE）计算 t 时刻所预测单词的概率分布 p_{t+1} 与正确单词的分布 y_{t+1} 之间的差距，这个差距即是 t 时刻损失函数的值。假设这一句话有 T 个单词（包括句子的起始符号 `BOS` 与结束符号 `EOS`），于是我们便可得到 T 个损失值，将这些损失值求和即可得到生成这一句话的损失值。若我们用 $\mathcal{C}$ 表示预测生成的句子，并且 $\mathcal{C} = \{y'_1, y'_2, \cdots, y'_T\}$。同时，用 $\mathcal{Y}$ 表示正确单词组成的句子，即 $\mathcal{Y} = \{y_0, y_1, \cdots, y_T\}$。那么以上过程的数学形式如下：

$$\mathcal{L}_{\mathrm{CE}} = -\log\ p(\mathcal{C}|\mathcal{Y}) = -\sum_{t=1}^{T} \log p(y_t|y_{t-1}). \tag{3.1}$$

当编解码模型处于测试推断阶段时，在 t 时刻，与训练阶段不同的是，我们是不知道这时刻正确的单词 y_t 的。但是如果要生成一句话，我们需要给 LSTM 单元提供一个单词词向量作为输入。

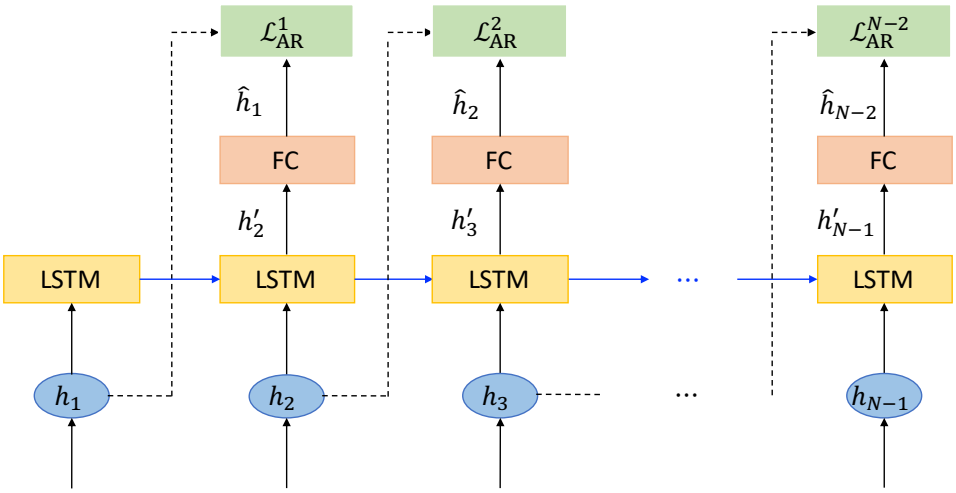

图 3.1: 自动重构网络结构示意图

这时候，我们用上一个时刻产生的单词 y'_t 所对应的词向量当作 t 时刻需要输入的单词。于是，测试推断阶段的过程就变成了这样，从句子的起始符号 BOS 开始，我们生成一个单词 y'_1，然后将这个单词输入到下一个时刻 LSTM 单元中来生成下一个单词 y'_2。依次类推，直到我们生成了句子的结束符号 EOS 之后，便结束生成过程，将生成的单词组成一句话。从这个过程中，便可发现，由于测试推断阶段我们不知道 t 时刻正确的单词 y_t，用的是上一个时刻产生的单词 y'_t 作为输入来生成 t 时刻的单词 y'_{t+1}。但是一旦这个过程中，有个单词生成的结果太差，那么便会引起连锁反应，即一步错，下面就是步步错了。

这种训练阶段与测试推断阶段两种模式之间的差异性问题，就是所谓的曝光偏差（exposure bias）问题，困扰着编解码图像语义描述模型。即便使用强化学习直接优化评测目标，如 BLEU、CIDEr 等等，但因为强化学习算法直接从随机初始化的参数开始训练会存在很大的偏差(variance)，导致模型训练很不稳定。因此，采用强化学习算法的图像语义描述模型也是先用交叉熵训练模型至收敛之后，在接着精调（fine-tuning）模型的。

3.2 自动重构网络的模型结构

为了应对上述编解码模型中存在的曝光偏差问题，本文提出了一种叫做自动重构网络的模型结构。它内嵌于编解码网络模型中，具体来说，这种结构是嵌入于序列解码网络之上。所谓的自动重构网络，是通过使用当前 t 时刻 LSTM 网络输出的隐状态 h_t 去重构出前一个时刻的隐状态 h_{t-1}。我们这样做的出发点是在 LSTM 中每个时刻的隐状态 h_t 都与它前一个时刻的隐状态息息相关，因为 $h_t = \text{LSTM}(x_t, h_{t-1})$。通过我们的自动重构网络，能挖掘并增强当前时刻的隐状态 h_t 与前一刻时刻的隐状态 h_{t-1} 之间更深层次的关系。

在本文中，如图3.1所示，自动重构网络实际上是采用另一层 LSTM 网络来实现的。我们用 h'_t 来表示 t 时刻时自动重构网络中 LSTM 循环体单元输出的隐状态。同时我们还将解码 LSTM 网

络的输出 h_t 输入进自动重构网络中。于是在 t 时刻，自动重构网络的 LSTM 中有两个输入，一个是解码 LSTM 网络 t 时刻输出的隐状态 h_t，另一个是自动重构 LSTM 网络的上一个时刻输出的隐状态 h'_{t-1}。于是，每一个时刻自动重构网络的内部有如下形式的变换：

$$\begin{aligned}\begin{pmatrix} i'_t \\ f'_t \\ o'_t \\ g'_t \end{pmatrix} &= \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \mathbf{T} \begin{pmatrix} h_t \\ h'_{t-1} \end{pmatrix}, \\ c'_t &= f'_t \odot c'_{t-1} + i'_t \odot g'_t, \\ h'_t &= o'_t \odot \tanh(c'_t).\end{aligned} \tag{3.2}$$

上式中，i'_t、f'_t、o'_t、c'_t 和 h'_t 分别为自动重构 LSTM 网络中的输入门、遗忘门、输出门、循环体记忆状态以及隐状态；$\odot$ 为逐元素点积运算符。同时自动重构网络输出的隐状态 h'_t 即为解码 LSTM 网络输出的 h_t 经过重构网络变换的一个结果。为了能更好的匹配解码 LSTM 网络前一个时刻的输出 h_{t-1}，我们还将 h'_t 经过一层全连接层进行线性变换，这样能更好的将 h'_t 映射到与 h_{t-1} 相近的空间。于是，我们有：

$$\hat{h}_{t-1} = W_{\text{fc}} h'_t + b_{\text{fc}}. \tag{3.3}$$

上式中，W_{fc} 和 b_{fc} 是全连接层的参数权重以及偏置项。$\hat{h}_{t-1}$ 即为 h_t 通过我们自动重构网络重构后的结果。之后，我们采用欧式距离（Euclidean distance）衡量 $\hat{h}_{t-1}$ 与解码 LSTM 网络前一个时刻真实隐状态 h_{t-1} 之间的距离：

$$\mathcal{L}^t_{\text{AR}} = \| h_{t-1} - \hat{h}_{t-1} \|_2^2 . \tag{3.4}$$

上式中，$\mathcal{L}^t_{\text{AR}}$ 代表了 t 时刻我们自动重构网络的重构误差。而我们的目标是缩小这个误差，以此达到两个目标，一是使得 t 时刻解码 LSTM 网络的隐状态 h_t 能够吸收并包含上一个时刻隐状态 h_{t-1} 中更多有用的信息，增强解码 LSTM 网络中相邻两个时刻隐状态之间的耦合性。二是通过我们的自动重构网络，能够充分的挖掘 h_t 与前一个时刻 h_{t-1} 之间更高层次语义关系，进一步增强前后两个时刻隐状态之间的联系。同时，在实验中我们发现，通过使用我们的自动重构网络，能够有效缓解前文所述的爆照偏差问题。这一点我们会通过下一章的定量化实验来验证，并通过 t-SNE [62] 可视化的技术直接展示我们自动重构网络在缓解曝光偏差问题上的显著效果。

需要注意的是，我们这里所提出的自动重构网络，用 h_t 通过自动重构网络去重构出 h_{t-1} 的思想，跟 Krueger [40] 等人提出的对循环神经网络中的状态进行正则化的 Zoneout 算法是既有相似性但又有区别。Zoneout 是将循环神经网络中的隐状态以及记忆状态以一定的概率不做更新，直接将数值复制到下一个时刻的 LSTM 单元中来。在使用 Zoneout 的 LSTM 循环体单元中，每一

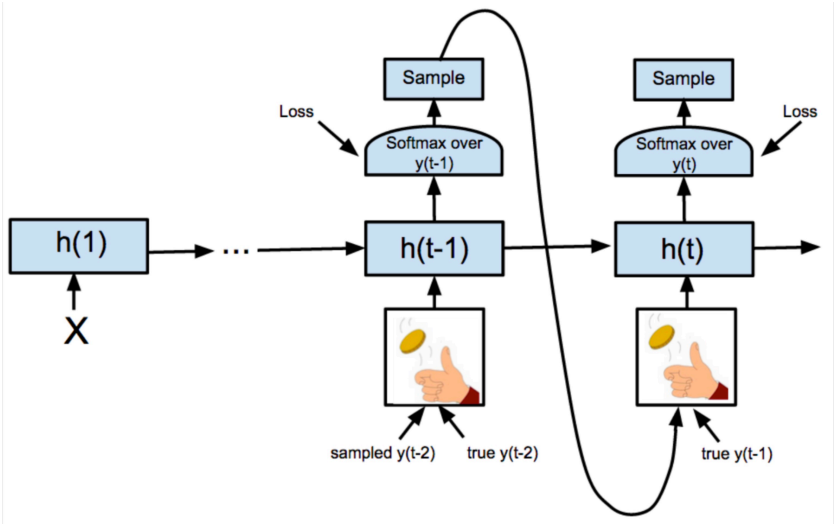

图 3.2: 解码网络中的 *Scheduled Sampling* [39] 算法示意图

时刻，其内部的记忆状态及隐状态的更新规则如下：

$$\begin{aligned} c_t &= d_t^c \odot c_{t-1} + (1 - d_t^c) \odot (f_t \odot c_{t-1} + i_t \odot g_t), \\ h_t &= d_t^h \odot h_{t-1} + (1 - d_t^h) \odot (o_t \odot \tanh(f_t \odot c_{t-1} + i_t \odot g_t)). \end{aligned} \tag{3.5}$$

上式中，d_t^c 是伯努利分布（Bernoulli distribution）随机采样得到的掩膜向量（与 LSTM 设置的隐状态元素个数相等）。这个掩膜向量中的元素值要么是 0，要么是 1；$\odot$ 是逐元素点积运算符。通过将前一个时刻 LSTM 网络单元中的隐状态、记忆状态随机保留到当前时刻的 LSTM 网络单元中，这样隐状态、记忆状态中所包含的信息能够随时间更加健全稳定的流动到后面时刻的 LSTM 网络单元中，从而起到提高循环神经网络建模能力的效果。但是，Zoneout 这种方法是一种生硬的策略，在循环神经网络中的每个时刻，通过伯努利采样，随机的选择是正常更新 LSTM 网络单元的内部状态还是复制上一个时刻 LSTM 网络单元的内部状态。这种正则化的方法，本质上可以看作由多个不同长度的 LSTM 网络所组成的集成模型（ensemble models）。而我们所提出的自动重构网络，它是一种软性的正则化方法，将 t 时刻的隐状态 h_t 经过自动重构网络，去重构出前一时刻的隐状态 h_{t-1}。这样不仅使得 h_t 能够从 h_{t-1} 中吸收并包含更多有用的信息，而且还能挖掘相邻隐状态之间的关系。所以，自动重构网络能够通过模型的训练自适应的决定当前时刻的隐状态和前一个时刻隐状态信息之间的联系，相比较于 Zoneout 跟据伯努利分布随机决定的方式，能够更好的利用好循环神经网络中隐状态中的信息。

如前文绪论所述，Bengio [39] 等人所提出的规定采样（Scheduled Sampling，SS）算法也能够正则化循环神经网络来提高序列生成的效果。如图3.2所示，所谓的 Scheduled Sampling 算法，是在解码 LSTM 网络的第 0 时刻，输入网络的还是正确的单词 y_0（实际上为语句起始符 `BOS`）。但从第 1 个时刻开始，通过随机采样，以一定的概率 ϵ^i，来决定当前时刻输入到网络里面的是正确的单词 y_t，还是采用上一时刻解码 LSTM 网络自己预测的单词 y_t'，这个概率则为 $1 - \epsilon^i$。虽然规定采样方法能够在训练阶段就以一定的概率随机模仿测试推断阶段的行为模式，但其实规定采样

是在训练阶段就将自己产生的数据加入到训练过程中，本质上是进行了训练数据的扩充。而我们所提出的模型是利用自动重构来充分挖掘相邻两个时刻隐状态之间更高层次语义的关系，增强它们之间信息的相关性、耦合性，从而提高序循环神经网络生成序列的效果。

3.3 自动重构网络的训练策略与讨论

我们的自动重构网络被嵌入于基础的编解码网络模型或带有注意力机制的编解码网络模型中，当加入使用自动重构网络之后，编解码网络模型的训练过程就由两个阶段组成。在第一个阶段，我们先将自动重构网络中的参数权重冻结，即不参与到编解码网络模型的训练中。同时，在这个阶段，用交叉熵损失函数正常的训练编解码网络模型，也就是采用如公式3.1所示的损失函数。当编解码网络模型收敛之后，我们进入第二个阶段。此时，我们“解冻”自动重构网络的参数，将自动重构网络的损失函数结合编解码网络模型的交叉熵损失继续精调整个模型。所以在第二个阶段，我们的损失函数由两部分所构成，一个是编解码网络模型的交叉熵损失函数，另一个则是我们的自动重构误差损失。具体的数学形式如下：

$$\mathcal{L} = \mathcal{L}_{\mathrm{CE}} + \lambda \sum_{t=1}^{T-1} \mathcal{L}_{\mathrm{AR}}^{t}. \tag{3.6}$$

上式中，超参数 λ 是自动重构损失的权重，用于控制自动重构所造成的损失对编解码网络模型的影响。所以设置一个合理的重构权重很重要，因为如果自动重构权重过大，则会导致解码网络中当前时刻的隐状态 h_t 与前一个时刻的隐状态 h_{t-1} 过于接近，这样会使得 h_t 过于吸收前一个时刻 h_{t-1} 中的信息，从而难以从当前时刻输入的 x_t 中吸收新的信息。反之，若自动重构权重设置过小，那么自动重构网络的所挖掘的 h_t 与 h_{t-1} 之间高层次语义联系难以对编解码网络模型施加影响。

之所以采用上述分阶段的训练策略来训练融入自动重构网络的编解码网络模型，是因为自动重构网络的目的是充分挖掘当前时刻 LSTM 网络输出的隐状态 h_t 与前一个时刻隐状态 h_{t-1} 之间更高层次的语义关系，同时鼓励 h_t 从 h_{t-1} 中吸收更多有用的信息。但这样做实际上是有一个前提条件的，就是我们的自动重构过程需要在 LSTM 的隐状态 h_t 能够对所输入的单词 y_t 正确编码的情况下，才能够发挥作用。如果在一开始就将自动重构网络加入到编解码网络模型的训练中，因为最开始的时候，整个网络模型的参数都是随机初始化的，LSTM 网络还不能够对输入的单词进行正确的编码，此时各个时刻的隐状态中所包含的信息都不正确，或者说实际上都是噪声。在这个时候如果就使用自动重构网络，即用 h_t 重构出 h_{t-1}，接着得到这个时刻的重构损失。但是，这样的损失计算实际上是没有意义的。如果此时再将这个损失函数的梯度通过反向传播算法去影响整个编解码网络模型的参数权重，那么整个网络模型容易陷入到一个不好的局部最优点。

我们所提出的自动重构网络也可以理解为对已收敛网络中相邻两个时刻之间隐状态关系的知识挖掘（knowledge distilling）。出发点如前文所述，相邻两个时刻隐状态之间的信息联系很紧密，

可以通过我们的自动重构网络来挖掘更高语义层次的关系，同时能够促使当前时刻的隐状态从前一个时刻的隐状态中吸收更多有用的信息，增加了相邻两个时刻隐状态之间信息的耦合性，从而达到缓解曝光偏差问题的目的。

4 实验结果和分析

这一章的主要内容为实验结果及分析。在第一节，我们先介绍图像语义描述任务所使用的标准数据集——MSCOCO [18]，然后介绍图像语义描述所使用的质量评测标准。在第二节中，详细介绍了我们在实现、测试模型时的各种参数配置。在第三节，我们通过在 MSCOCO 数据集上的一系列实验与分析，来验证前文所介绍的一般编解码模型（Encoder-Decoder），以及带有注意力机制的编解码模型（Attentive Encoder-Decoder）的性能。并且，我们也会将前文所提出的自动重构网络（ARNet）嵌入到一般的编解码模型以及带有注意力机制的编解码模型中，以此验证我们提出的自动重构网络对于模型性能的帮助与提升。同时，为了与前文所述的两种循环神经网络正则化方法：规定采样（Scheduled Sampling）以及 Zoneout 进行比较，我们实现并在数据集上测试验证了上述两种算法，并将我们的自动重构网络与这两种算法进行比较分析。在第四节，我们量化分析了自动重构网络对于缓解编解码模型中曝光偏差问题的效果。随后的第五节，我们进一步通过实验，讨论了不同大小的自动重构权重对于模型性能的影响。最后，我们将自动重构网络用于置换顺序的序列化 MNIST 手写数字分类任务，说明自动重构网络不仅适用于图像语义描述中的解码网络，还适用于使用循环神经网络对输入数据进行编码的任务。

4.1 实验数据集和评测标准

4.1.1 MSCOCO 数据集

MSCOCO 数据集是继 ImageNet 之后规模最大，使用范围最广的图像数据集之一。它为各种各样的视觉任务提供了全面、细致的标注数据，它可以用于通用物体检测（object detection），关键点检测（keypoint detection），场景语义解析（Scene Semantic Parsing），以及我们这里的图像语义描述（image semantic captioning）任务。近些年来，所有的图像语义描述的工作都是基于 MSCOCO 数据集来进行比较的。

离线测试（offline test）：对于图像语义描述任务，MSCOCO 里包含多达 123,000 张图像，为每一张图像提供了等于或者大于 5 个句子的描述语句。由于 MSCOCO 公布的数据中，对于测试集，只提供了图像，没有提供正确的标注语句。因此，Karpathy [63] 等人从官方提供的验证集里分割出 5000 张图像作为验证集、5000 张作为测试集，剩余的 30,000 多张图像与原先的 80,000 张训练图像合并组成新的训练集。用这样分割的数据集进行实验的成为离线测试（offline）。后来的研究者为了将模型的测试结果进行公平的比较，均采用 Karpathy 分割方式的数据进行实验与评测。

在线测试(online test)：如果按照 MSCOCO 官方的数据集分割方式进行训练模型，将训练好的模型在官方的四万多张测试图像上进行测试，生成测试图像的语义描述之后，上传到 MSCOCO 的官方评测网站进行评测，从而得到模型在官方测试集上的测试得分，这样的训练测试模型的方式称作在线测试（online）。

4.1.2 图像语义描述评价标准

将一张图像输入到模型中，模型生成改图像的语义描述语句，我们用一系列自动评价标准（automatic evaluation metric）来评判所生成语句的质量。目前图像语义描述的评价标准有很多，如 BLEU、METEOR、ROUGE-L 及 CIDEr。我们下面一一介绍：

BLEU [29]： BLEU 的全称为 Bilingual Evaluation Understudy。BLEU 最开始是被用来评价机器翻译效果的，通过分析模型生成的翻译语句和正确的翻译语句之间单词 n 元组的相关性来进行评价，它属于统计评价标准。后来也被用于对图像语义描述任务的生成结果进行评价。对于输入图像 I，图像语义描述模型生成的描述语句为 $\mathcal{C}_I = \{y'_1, y'_2, \cdots, y'_T\}$。同时图像 I 也有 m 句参考描述语句，也就是人工标注的图像语义描述语句，记为 $\mathcal{Y}_I = \{\mathcal{Y}_I^1, \mathcal{Y}_I^2, \cdots, \mathcal{Y}_I^m\}$，其中 $\mathcal{Y}_I^m = \{y_1^m, y_2^m, \cdots, y_T^m\}$。在用 BLEU 评价标准进行评价时，图像语义描述语句都是用 n 元组（n-gram）表示的，一个 n 元组记为 w_k，它是由 n 个有顺序单词所组成的连续序列。在图像语义描述任务中，我们一般用一元组、二元组、三元组直至四元组来评价图像描述语句，分别记为 BLEU-1、BLEU-2、BLEU-3 及 BLEU-4。n 元组 w_k 在语句参考描述语句 $\mathcal{Y}_I^j$ 中出现的次数被记为 $s_k(\mathcal{Y}_I^j)$。n 元组 w_k 在模型生成的待评价语句 $\mathcal{C}_I$ 中出现的次数记为 $s_k(\mathcal{C}_I)$。于是，明确上述符号含义后，先计算待评价描述语句 $\mathcal{C}_I$ 在 m 句参考描述语句 $\mathcal{Y}_I$ 中全局 n 元组的精度 $p_n(\mathcal{C}_I, \mathcal{Y}_I)$，计算方式如下式：

$$p_n(\mathcal{C}_I, \mathcal{Y}_I) = \frac{\sum_k \min\left(s_k(\mathcal{C}_I), \max_{j \in m} s_k(\mathcal{Y}_I^j)\right)}{\sum_k s_k(\mathcal{C}_I)}. \tag{4.1}$$

其次，我们还需要计算简洁性惩罚值（brevity penalty），用 $bp(\mathcal{C}_I, \mathcal{Y}_I)$ 表示：

$$bp(\mathcal{C}_I, \mathcal{Y}_I) = \begin{cases} 1, & \text{if } l_{\mathcal{C}} > l_{\mathcal{Y}}, \\ e^{1-l_{\mathcal{Y}}/l_{\mathcal{C}}}, & \text{if } l_{\mathcal{C}} \leq l_{\mathcal{Y}}. \end{cases} \tag{4.2}$$

其中，$l_{\mathcal{C}}$ 是待评价描述语句 $\mathcal{C}_I$ 的总长度，$l_{\mathcal{Y}}$ 是图像 I 所有参考句子 $\mathcal{Y}_I$ 的总长度。于是，便可计算图像 I 所生成的描述语句 $\mathcal{C}_I$ 的在元组数为 N 时的 BLEU 分数：

$$\text{BLEU}_N(\mathcal{C}_I, \mathcal{Y}_I) = bp(\mathcal{C}_I, \mathcal{Y}_I)\, exp\left(\sum_{n=1}^{N} w_n log\left(p_n(\mathcal{C}_I, \mathcal{Y}_I)\right)\right) \tag{4.3}$$

ROUGE-L [32]： ROUGE-L 是 ROUGE 评判标准中的其中一个评判标准，实际上 ROUGE 共有 3 个评价标准，分别是 ROUGE-N，ROUGE-L 以及 ROUGE-S。它是原本被用来评价文本摘要算法的自动评价标准集，我们这里将其中的 ROUGE-L 作为图像语义描述任务的评价标准。

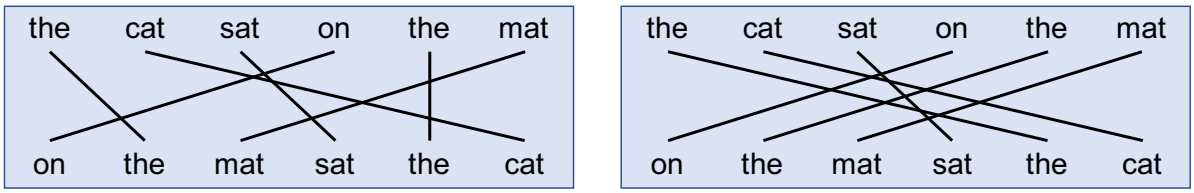

图 4.1: *METEOR* 中对齐映射集（*alignments*）示意图

ROUGE-L 是一种基于最长公共子序列（longest common subsequence，LCS）的评价方法。公共子序列原本是指多个字符串之间可具有的长度最大的公共子序列，在这里将描述语句看作字符串，将组成描述语句的单词看作一个个字符。用 $l\left(\mathcal{C}_I, \mathcal{S}_I^j\right)$ 表示对图像 I 模型所生成的描述语句和人工标注的参考语句之间最长公共子序列的长度，其中 $j \in [1, m]$。于是，ROUGE-L 的评价得分可由下式计算：

$$\begin{aligned} R_l &= \max_j \frac{l\left(\mathcal{C}_I, \mathcal{S}_I^j\right)}{\left|\mathcal{S}_I^j\right|}, \\ P_l &= \max_j \frac{l\left(\mathcal{C}_I, \mathcal{S}_I^j\right)}{\left|\mathcal{C}_I\right|}, \\ \text{ROUGE-L}\left(\mathcal{C}_I, \mathcal{S}_I\right) &= \frac{(1+\beta^2) R_l P_l}{R_l + \beta^2 P_l}. \end{aligned} \tag{4.4}$$

上式中，R_l、P_l 分别为召回率、精度，β 一般等于 1.2。

METEOR [30]：METEOR 是被设计用来解决 BLEU 评价标准的缺点的，它考虑了待评价的描述语句 $\mathcal{C}_I$ 和人工标注的参考语句 $\mathcal{S}_I^j$ 之间的对齐关系，它和人工判断的结果有更高的相关性。METEOR 先在两个句子之间创建一个对齐映射集（alignments），即两个句子之间一元组的映射集。具体来说，需要将待评价描述语句中的每个一元组需要映射到参考标注语句中的一个或零个一元组。对于待评价描述语句 $\mathcal{C}_I$ 与多个参考描述语句有对齐映射集，且它们的映射数量相同，那么选择映射交叉数目最少的，如图4.1所示，我们选择左图作为映射匹配结果。之后便计算 METEOR 得分，先计算召回率及精度：

$$\begin{aligned} R_m &= \frac{m}{w_{\mathcal{C}_I}}, \\ P_m &= \frac{m}{w_{\mathcal{S}_I^j}}, \end{aligned} \tag{4.5}$$

其中 m 是在参考图像描述语句中同样存在的，待评价描述语句中一元组的数量。$w_{\mathcal{C}_I}$ 是待评价描述语句中的一元组数量。$w_{\mathcal{S}_I^j}$ 是参考图像描述语句中一元组的数量。然后使用调和平均（harmonic-mean）来计算 $Fmean$，且召回率 R_m 的权重是精度 P_m 的 9 倍。

$$Fmean = \frac{10 P_m R_m}{R_m + 9 P_m}, \tag{4.6}$$

以上的精度 P_m、召回率 R_m 以及 $Fmean$ 都是基于一元组来匹配计算的，只对生成句子中单个单词的一致性进行了评价，却没有对参考描述语句和待评价描述语句中更长的多元组的一致性进行

评价。因此，需要对一元组生成的对齐映射集计算一个惩罚值。即若参考描述语句和待评价描述语句中没有相邻的映射越多，则惩罚值就越高，这里相邻的映射的多元组被定义为块（chunks）。如有这样两句话，生成的语句为 “the predident spoke to the audience”，以及参考描述语句为 “the predident then spoke to the audience”。这里面就包括了两个所谓的 “块”，一个是词组 “the predident”，另一个是词组 “spoke to the audience”。于是惩罚项（$Penalty$）的值计算如下：

$$Penalty = 0.5 * \left(\frac{\#chunks}{\#unigrams\_matched} \right)^3, \tag{4.7}$$

上式中，$\#chunks$ 是块的数量，$\#unigrams\_matched$ 是被映射的一元组的数量。最后，METEOR 值计算如下：

$$\text{METEOR}\left(\mathcal{C}_I, \mathcal{S}_I\right) = Fmean * (1 - Penalty). \tag{4.8}$$

CIDEr [31]：CIDEr 评价指标是专门设计用来对图像语义描述任务的进行评价的。它是通过对每个 n 元组计算词频-逆文档频率（Term Frequency Inverse Document Frequency，TF-IDF）作为权重，来衡量图像描述语句之间的一致性。对于一张图像 I，它是训练图像数据集 $\mathcal{I}$ 中的一张图像。一个 n 元组 w_k 在参考描述语句 $\mathcal{S}_I^j$ 中的次数记为 $h_k\left(S_I^j\right)$，出现在生成的待评价语句中的次数记为 $h_k\left(\mathcal{C}_I\right)$。那么每个 n 元组 w_k 的 TF-IDF 权重 $g_k\left(S_I^j\right)$，计算方式如下：

$$g_k\left(\mathcal{S}_I^j\right) = \frac{h_k\left(\mathcal{S}_I^j\right)}{\sum_{\omega_l \in \Omega} h_l\left(\mathcal{S}_I^j\right)} \log\left(\frac{|\mathcal{I}|}{\sum_{I_p \in \mathcal{I}} \min\left(1, \sum_q h_k\left(\mathcal{S}_p^q\right)\right)} \right). \tag{4.9}$$

上式中，Ω 是所有 n 元组的词汇表，$\mathcal{I}$ 是数据集中所有图像的集合。上面公式中第一项计算的是每个 n 元组 w_k 的词频（Term Frequency，TF），即 w_k 这个词出现的频次；第二项计算的是 n 元组 w_k 的逆文档频率（Inverse Document Frequency，IDF），即这个词在当前整个文档中的稀有程度，这里我们将这个数据集所有的人工标注语句当作整个文档。如果一个 n 元组频繁的出现在图像的参考描述语句中，TF 对于这些 n 元组将给出更高的权重，而 IDF 则降低那些在所有描述语句中都常常出现的 n 元组的权重。也就是说，IDF 提供了一种测量单词显著性的方法，这就是将那些容易常常出现，但是对于视觉内容信息没有多大帮助的单词的重要性打折扣。对于长度为 n 的 n 元组的 CIDEr_n 分数是使用待评价描述语句和人工标注的参考描述语句之间的平均相似性来计算：

$$\text{CIDEr}_n\left(\mathcal{C}_I, \mathcal{S}_I\right) = \frac{1}{m} \sum_j \frac{g^n(\mathcal{C}_I) \cdot g^n(\mathcal{S}_I^j)}{\left\|g^n(\mathcal{C}_I)\right\| * \left\|g^n(\mathcal{S}_I^j)\right\|}. \tag{4.10}$$

于是，综合不同 n 值的 n 元组的得分，总的 CIDEr 得分计算如下：

$$\text{CIDEr}\left(\mathcal{C}_I, \mathcal{S}_I\right) = \sum_{n=1}^{N} w_n \text{CIDEr}_n\left(\mathcal{C}_I, \mathcal{S}_I\right). \tag{4.11}$$

上式中，w_n 的值一般取 $w_n = \frac{1}{N}$，我们一般取 $N = 4$，即最高考虑到四元组即可。

4.2 实验的参数配置

我们先对 MSCOCO 官方提供的由人工标注的图像语义描述语句进行预处理。具体来说，先将描述语句中所有的大写字母转成小写，移除掉描述语句中所有不是字母或者数字的字符，如美元符号等等。之后从这些描述语句中提取并建立 MSCOCO 数据集的词汇表以及每个单词出现的频次，过滤掉出现次数小于 5 次的单词，最终的词汇表中包含有 10,516 个单词。考虑到训练时间效率，以及 LSTM 循环神经网络的记忆能力，我们设置描述语句的最大长度为 30（包括句子的起始符 `BOS` 和终止符 `EOS`），单词数长于 30 个单词的语句我们会将其截断。而单词数少于 30 个单词的句子，我们则用 0 补齐到 30 个长度。所有的描述语句都以起始符 `BOS` 开始，以终止符 `EOS` 结束。

在我们的模型实现时，采用在 ImageNet 上预训练好的模型 Inception-v4 作为图像特征编码网络。我们将 Incepton-v4 图像编码网络中平均池化层（Average Pooling）的输出作为输入图像的全局特征，记为 g。同时提取 Inception-v4 图像编码网络中最后一个 Inception-C 模块输出的一系列特征图作为图像的局部特征表达，记为 $\mathbf{s}$。更具体的，我们得到的图像全局特征表示 g 的维度是 1536，局部特征表达 $\mathbf{s} = \{s_1, ..., s_{64}\}$ 则是由 64 个 1536 维的向量组成。在编解码模型的整个训练阶段，为了加速训练过程同时为了防止编码网络过拟合，我们将图像编码网络的参数权重固定起来，不加入整个网络的训练。对于解码网络和自动重构网络中所使用的 LSTM 循环神经网络，隐状态参数的值设置为 512。对于将单词映射为词向量的词嵌入网络，我们将词向量的维度同样设置为 512。

如前文所述，结合我们提出的自动重构网络的编解码网络模型的训练过程分为两个阶段，第一个阶段是以公式3.1为损失函数训练编解码网络模型，直至基础的模型收敛。接着在第二个阶段，将自动重构网络加入到训练中，进一步精调网络，这时候的损失函数为公式3.6，同样训练至收敛。在整个训练过程中，我们均使用 Adam [64] 梯度下降优化器来优化目标函数。在第一个阶段，我们选取的初始学习率为 5×10^{-4}。在对模型进行精调的第二个阶段，我们选取的学习率为 1×10^{-4}。在每个训练阶段中，学习率的衰减策略均是每训练 3 轮（epoch），学习率会在原来的基础上乘以 0.8。这里的每一轮（epoch）的意思是指训练过程完整的过了一遍训练集中所有的样本。每一次开始新的一轮时，我们会将训练集中的图像顺序都随机打乱，以防止模型的过拟合。训练中每一次迭代的样本数量（batch size）均为 80。

为了防止在模型训练阶段，模型陷入过拟合的状态，我们使用了提早停止（Early Stopping）的训练策略。具体来说，当在训练集上每一轮（epoch）训练迭代完成之后，我们将此刻的模型在验证集上进行测试，对测试集生成的图像语义描述语句用 CIDEr 指标进行评价，得到一个 CIDEr 指标的分数。将当前 epoch 模型的 CIDEr 值与之前模型最高的 CIDEr 值进行比较，如果有提升，我们暂时将当前 epoch 训练完成的模型参数作为最终的模型参数，同时继续进行训练。如果连续 5 个 epoch 训练完成的模型，在验证集上的 CIDEr 值相对于之前都不再有提高，我们则停止训练，

表 4.1: 各种方法的单个模型在离线 *MSCOCO* 数据集上的性能比较

模型名称	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	ROUGE-L	CIDEr
DVSA [63]	0.625	0.450	0.321	0.230	0.195	-	0.660
NIC [10]	0.663	0.423	0.277	0.183	0.237	-	0.855
m-RNN [15]	0.600	0.410	0.280	0.190	0.228	-	0.842
Soft-Attention [16]	0.707	0.492	0.344	0.243	0.239	-	-
Hard-Attention [16]	0.718	0.504	0.357	0.250	0.230	-	-
Semantic Attention [19]	0.709	0.537	0.402	0.304	0.243	-	-
Review Net [50]	-	-	-	0.290	0.237	-	0.886
LSTM-A5 [27]	0.730	0.565	0.429	0.325	0.251	0.538	0.986
Text-guided Attention [28]	**0.749**	**0.581**	0.437	0.326	0.257	-	1.024
Encoder-Decoder	0.718	0.547	0.412	0.311	0.251	0.530	0.961
Encoder-Decoder + Zoneout	0.708	0.537	0.403	0.304	0.249	0.525	0.941
Encoder-Decoder + Scheduled Sampling	0.718	0.548	0.414	0.315	0.252	0.531	0.975
Encoder-Decoder + ARNet	0.730	0.562	0.425	0.321	0.252	0.535	0.988
Attentive Encoder-Decoder	0.727	0.557	0.421	0.318	0.259	0.537	0.996
Attentive Encoder-Decoder + Zoneout	0.720	0.549	0.415	0.314	0.251	0.532	0.975
Attentive Encoder-Decoder + Scheduled Sampling	0.731	0.563	0.426	0.322	0.256	0.538	1.006
Attentive Encoder-Decoder + ARNet	0.740	0.576	**0.440**	**0.335**	**0.261**	**0.546**	**1.034**

同时我们选取之前 CIDEr 值最大的 epoch 所训练完成的模型参数作为最终的模型参数。这里，之所以选择 CIDEr 值作为评判指标，是因为 CIDEr 是当前图像语义描述任务中相对于其他几个评价标准而言是最好的评判指标，同时 MSCOCO 排行榜上也默认将 CIDEr 分数的排名作为最终的排名。

4.3 模型的性能比较与分析

为了公平的比较各个模型生成的图像描述语句的准确性、质量，我们使用的是官方提供的图像语义描述评价工具[1]，它实现并集成了上述的各种评价标准，包括 BLEU@N、METEOR、ROUGE-L 和 CIDEr，以便图像语义描述的研究者将他们的模型方法更加公平的比较。本文将 NIC [10] 编解码网络模型作为本研究中图像语义描述的基础模型（Encoder-Decoder）。将基础的编解码网络结构结合上 Xu [16] 等人提出的软性注意力机制，作为带有注意力机制的编解码网络（Attentive Encoder-Decoder）。我们的自动重构网络（ARNet）分别嵌入到基础的编解码网络以及带有注意力机制的编解码网络中，同时将它们分别表示为 Encoder-Decoder-ARNet，以及 Attentive Encoder-Decoder-ARNet。为了与前文所提到的另外两种循环神经网络正则化的方法：Scheduled Sampling 以及 Zoneout 进行比较，我们也将这两种正则化的方法也分别融合到 Encoder-Decoder 以及 Attentive Encoder-Decoder 两种模型框架中。本文也将我们的方法与很多前人具有代表性的工作进行比较，如 m-RNN [15]、Semantic Attention [19]、Review Net [50]、LSTM-A5 [27] 以及 Text-guided Attention [28] 这些方法。

[1] https://github.com/tylin/coco-caption

表 4.2: 各种不同的模型在在线 *MSCOCO*（*C-5*）数据集上的性能比较

模型名称	C@N	BLEU@1	BLEU@2	BLEU@3	BLEU@4	METEOR	ROUGE-L	CIDEr
NIC [10]	C-5	0.713	0.542	0.407	0.309	0.254	0.530	0.943
m-RNN [15]	C-5	0.716	0.545	0.404	0.299	0.242	0.521	0.917
Soft/Hard Attention [16]	C-5	0.705	0.528	0.383	0.277	0.241	0.516	0.865
Semantic Attention [19]	C-5	0.731	0.565	0.424	0.316	0.250	0.535	0.943
Review Net [50]	C-5	-	-	-	-	-	-	-
LSTM-A5 [27]	C-5	0.739	0.575	0.436	0.330	0.256	0.542	0.984
Text-guided Attention [28]	C-5	**0.743**	0.575	0.431	0.321	0.255	-	0.987
Attentive Encoder-Decoder + ARNet	C-5	0.741	**0.576**	**0.438**	**0.332**	**0.259**	**0.544**	**1.006**

表4.1展示了在离线版本的 MSCOCO 数据集上，不同方法模型的性能比较。从表中我们可以看出，无论是对于基础的编解码网络还是对于带有注意力机制的编解码网络，加入我们的自动重构网络后，在各个评价标准上均能提高模型的性能。相比较于另外两种对循环神经网络正则化的方法——Scheduled Sampling 以及 Zoneout，我们的自动重构网络亦能超过这两个方法。同时，普通的带有注意力机制的编解码模型在加入我们的自动重构网络后，在绝大部分的评价标准上，都超过了现今已发表的论文中所汇报的结果，达到了较为前沿的性能。虽然其中 BLEU-1 以及 BLEU-2 的评价得分略低于 Text-guided Attention [28] 的方法，是因为后者通过图像相似性从人工标注的图像描述中搜索相近的描述语句来辅助生成描述语句，相当于加入了图像的属性信息，从而使得 BLEU-1 和 BLEU-2 这样的评价指标显得较高。

表4.2以及表4.3展示了在在线版本的 MSCOCO 数据集上，各个模型性能的比较结果。其中，C-5 表示在测试数据集中，每张图像有 5 张人工标注的参考描述语句，而 C-40 则代表每张图像的人工标注参考语句从 5 句话扩展到了 40 句话。从表4.2中我们可以发现，使用自动重构网进行精调之后，带有注意力机制的模型生成的描述语句质量达到了最好的水平，只有在 BLEU-1 评价指标上弱于 Text-guided Attention [28] 的模型。在表4.3中，我们的模型在 CIDEr 以及 METEOR 这两个更加注重语句语义的指标上取得优势。说明使用当前时刻 LSTM 循环单元的隐状态 h_t 去重构前一时刻的隐状态 h_{t-1}，能够促使 h_t 从 h_{t-1} 吸收更多有用的信息，同时 h_t 与 h_{t-1} 之间更高层次的语义关系能够被我们的自动重构网络挖掘出来，从而提升描述语句语义的生成质量。虽然其余的几个指标，如 BLEU、METEOR，目前弱于 LSTM-A5 模型。原因是我们模型训练时更注重语义评价指标 CIDEr，使用提早停止（Early Stopping）策略时，也是根据 CIDEr 的值来选择模型参数权重的。但目前考虑到图像语义描述任务在 MSCOCO 数据集上是以 CIDEr 为首要评价指标，所以可以说我们的模型是达到前沿效果的。

图4.2展示了带有注意力机制编解码网络生成的图像语义描述结果，以及融入本文的自动重构网络进行精调之后模型生成的结果。从图中我们可以发现，我们提出的自动重构网络编解码模型能够生成更详细、更生动的图像语义描述语句。图中的一些单词或短语，如 “`keyboard`”、“`flowers`”、“`lush green hillside`”，表明了自动重构网络对解码网络生成语句的帮助。

表 4.3: 各种不同的模型在在线 *MSCOCO*（*C-40*）数据集上的性能比较

模型名称	C@N	BLEU@1	BLEU@2	BLEU@3	BLEU@4	METEOR	ROUGE-L	CIDEr
NIC [10]	C-40	0.895	0.802	0.694	0.587	0.346	0.682	0.946
m-RNN [15]	C-40	0.890	0.798	0.687	0.575	0.325	0.666	0.935
Soft/Hard Attention [16]	C-40	0.881	0.779	0.658	0.537	0.322	0.654	0.893
Semantic Attention [19]	C-40	0.900	0.815	0.709	0.599	0.335	0.682	0.958
Review Net [50]	C-40	-	-	-	0.597	0.347	0.686	0.969
LSTM-A5 [27]	C-40	**0.919**	**0.842**	**0.740**	**0.632**	0.350	**0.700**	1.003
Text-guided Attention [28]	C-40	0.915	0.832	0.722	0.607	0.341	-	1.001
Attentive Encoder-Decoder + ARNet	C-40	0.917	0.838	0.736	0.629	**0.351**	0.699	**1.017**

图像	生成的语义标注	正确的语义标注
	**Attentive Encoder-Decoder** a close up of a cat on a desk. **Attentive Encoder-Decoder-ARNet** a cat sitting on a desk next to a keyboard.	1. a grey cat peers at a computer keyboard. 2. a cat laying down by a keyboard. 3. a kitty playing with the keyboard on a laptop. 4. a large cat laying atop a computer keyboard. 5. a cat that is laying on a computer keyboard.
	**Attentive Encoder-Decoder** a display of many different types of cake. **Attentive Encoder-Decoder-ARNet** a cake decorated with many different types of red flowers.	1. a layered cake with many decorations on a table. 2. a large multi layered cake with candles sticking out of it. 3. a party decoration containing flowers, flags, and candles. 4. a cake decorated with flowers and flags on it. 5. a cake is decorated with flowers and flags.
	**Attentive Encoder-Decoder** a brown dog holding a blue frisbee in it's mouth. **Attentive Encoder-Decoder-ARNet** a dog running in the grass with a frisbee in it's mouth.	1. a very cute brown dog with a disc in its mouth. 2. a dog running in the grass with a frisbee in his mouth. 3. a dog in a grassy field carrying a frisbee. 4. a brown dog walking across a green field with a frisbee in it's mouth. 5. a dog carrying a frisbee in its mouth running on a grass lawn.
	**Attentive Encoder-Decoder** a truck driving down a road next to a forest. **Attentive Encoder-Decoder-ARNet** a car driving down a road next to a lush green hillside.	1. a street scene of a road going through the mountains. 2. a road curving around hills has one car on it. 3. a yellow car driving away on the road. 4. a small yellow and black car driving around the bend of a road between. 5. a small yellow car going around a turn and a sign.
	**Attentive Encoder-Decoder** a bus that is sitting in the street. **Attentive Encoder-Decoder-ARNet** a white bus driving down a street next to a building.	1. a black and white bus some bushes and building. 2. a white decorated bus is next to a building. 3. a large white bus that is by a building. 4. a large bus parked in a parking lot. 5. a white bus driving past a tall building.

图 4.2: 带有注意力机制的编解码网络模型以及融合自动重构网络后模型的图像语义描述实例

4.4 训练阶段与测试阶段差异性分析

正如3.1所分析的，基于编解码网络的图像语义描述模型存在训练阶段与测试推断阶段生成单词方式不一致的问题。在训练阶段的 t 时刻，解码网络根据 $t-1$ 时刻 LSTM 输出的隐状态 h_{t-1} 以及当前 t 时刻人工标注的单词 y_t，生成单词 y'_{t+1}。我们的训练目标是使得这个生成的单词 y'_{t+1} 的词汇表分布与人工标注的单词 y_{t+1} 的分布尽量“相近”，这就是所谓的最大似然估计。但是在推断测试阶段，t 时刻，人工标注的单词 y_t 我们是不知道的，于是我们用 $t-1$ 时刻生成的单词 y'_t 代替 y_t 输入到 LSTM 循环体单元中。于是，测试推断阶段解码网络容易产生误差并且不断地累

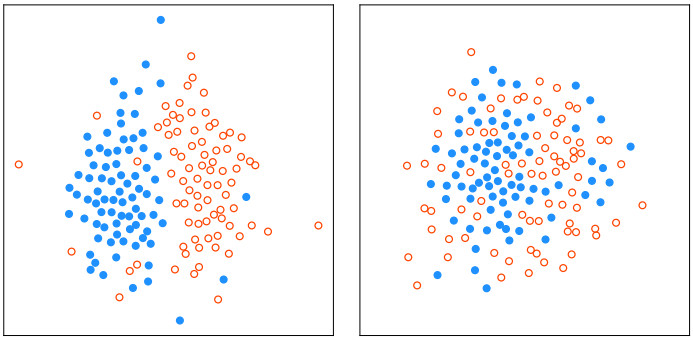

图 4.3: 带有注意力机制的编解码网络隐状态的分布和融入自动重构网络后模型隐状态的分布示意图

积，造成所谓的曝光偏差问题。为了定量化的研究这个问题，我们将序列解码网络 LSTM 循环体最后输出的隐状态当作生成句子的整体特征表达，这个隐状态的分布状态也即等同于这个序列语句的分布状态。

具体来说，我们将解码网络输出语句终止符 `EOS` 前一个时刻的隐状态 h_t 当作整个句子的特征表达。我们借助于 t-SNE [62] 可视化技术，将隐状态 h_t 的维度从 512 维降低到 2 维，这样可以方便的用二维图像展示隐状态的分布。图4.3中的左图与右图分别展示了带有注意力机制的编解码网络模型的隐状态的分布，以及融入我们的自动重构网络之后模型的隐状态的分布。其中，蓝色的实心圆点代表了在训练模式下提取的隐状态，空心的红色圆点代表了在测试推断模式下提取的隐状态。从图中我们可以看到，在未融入自动重构网络时，训练模式下提取的句子特征与测试推断模式下提取的句子特征状态分布分开的很明显，说明训练模式下的隐状态与测试推断模式下的隐状态在高维空间分布上分离的较远，也进一步说明了曝光偏差问题的存在。而经过我们的自动重构网络精调之后，我们可以发现，这两种模式下所提取的句子特征分布融合在一起了。从而说明了我们的自动重构网络对于缓解曝光偏差问题的有效性。

为了定量的研究编解码网络中模型的两种运行模式所造成的曝光偏差问题，我们需要设计一个距离度量指标。考虑到隐状态的模（$\|h_t\|$）大小的影响，欧式距离并不适合直接衡量两种高维向量的距离。因此，我们将余弦距离（cosine distance）作为距离度量指标。两个隐状态 h_1 和 h_2 之间的余弦距离计算如下：

$$d(h_1, h_2) = \frac{h_1^T h_2}{\|h_1\|\|h_2\|}. \tag{4.12}$$

余弦距离考虑到了隐状态 h_1 和 h_2 之间的角度距离，可以消除隐状态模的影响。

基于余弦距离，我们又定义了两种距离计算方法来衡量两种模式下所提取出的序列语句特征的距离。具体来说，定义 $\mathbf{U} = \{u_{\mathrm{I}_1}, ..., u_{\mathrm{I}_B}\}$ 为训练模式下提取到的解码网络的隐状态，定义 $\mathbf{V} = \{v_{\mathrm{I}_1}, ..., v_{\mathrm{I}_B}\}$ 为测试推断模式下提取到的解码网络的隐状态。我们定义的第一种距离度量方式是两种模式下提取的到的隐状态各自中心点之间的余弦距离，我们称之为平均中心距离（mean centroid

表 4.4: 不同的模型在训练模式与测试推断模式下隐状态之间的平均中心距离与逐元素距离

模型名称	平均中心距离 d_{mc}	逐元素距离 d_{pw}
Encoder-Decoder	0.747	0.719
Encoder-Decoder + Zoneout	0.782	0.755
Encoder-Decoder-Scheduled Sampling	0.713	0.700
Encoder-Decoder + ARNet	0.514	0.561
Attentive Encoder-Decoder	0.773	0.760
Attentive Encoder-Decoder + Zoneout	0.745	0.756
Attentive Encoder-Decoder + Scheduled Sampling	0.641	0.639
Attentive Encoder-Decoder + ARNet	**0.491**	**0.595**

distance)，记为 d_{mc}，则计算方式如下：

$$d_{\mathrm{mc}}\left(\mathbf{U},\mathbf{V}\right)=d\left(\frac{1}{B}\sum_{i}^{B}u_{\mathrm{I}_i},\frac{1}{B}\sum_{j}^{B}v_{\mathrm{I}_i}\right). \tag{4.13}$$

第二种距离度量方式是计算模型的两种模式下提取到的隐状态每个元素之间的距离，我们称为逐元素距离（point-wise distance)，记为 d_{pw}，计算方式如下：

$$d_{\mathrm{pw}}\left(\mathbf{U},\mathbf{V}\right)=\frac{1}{B}\sum_{i=1}^{B}d\left(u_{\mathrm{I}_i},v_{\mathrm{I}_i}\right). \tag{4.14}$$

也就是说，第二种距离度量方式 d_{pw} 只考虑相同图像在两种模式下得到的隐状态之间的余弦距离。

表4.4展示了带有注意力机制的编解码网络模型在两种模式下提取到的隐状态之间的平均中心距离和逐元素距离，以及融入自动重构网络之后模型的隐状态之间的两种距离值。从表中我们可以清楚的看到，经过我们的自动重构网络精调之后，模型训练模式的隐状态和测试推断模式下的隐状态之间的距离明显缩小。同时，我们的自动重构网络相比较于 Scheduled Sampling 正则化方法以及 Zoneout 正则化方法，更能有效的缓解曝光偏差问题。原因是在加入我们自动重构网络之后，相邻隐状态之间耦合性更高，每个时刻的隐状态 h_t 都能从前一个时刻的隐状态中吸收更多有用的信息。其次，隐状态之间高层次的语义特征也能够被自动重构网络挖掘出来。

4.5 重构网络权重的影响

因为我们的自动重构网络涉及到一个超参数（hyperparameter)，即自动重构的损失对于模型梯度更新影响的大小是通过一个超参数 λ 来控制的。如果 λ 的值设置为 0，那么整个网络模型又退回到普通的编解码网络模型。我们通过设置不同大小的 λ 值，来探究不同大小的自动重构损失对于模型性能的影响。图4.4展示了不同大小的权重值对带有注意力机制的编解码网络模型性能的

表 4.5: 多种模型在置换顺序的序列化 *MNIST* 手写数字分类上的性能比较

模型名称	测试精度
LSTM + dropout	0.925
LSTM + zoneout	0.931
Unregularized LSTM	0.914
LSTM + ARNet	**0.933**

影响（我们用 CIDEr 评价指标去衡量）。同样证明了在适当范围内，在使用我们的自动重构网络之后，能够提高编解码网络模型的性能。而太大的自动重构权重，模型的性能则会下降。这是因为过大的重构权重会导致模型只顾着去 “回忆” 之前时刻的信息，而忘记去吸收此时输入的信息，这样反而使得效果变差。在我们的实验中，我们发现 0.01 能够取得较好的效果。

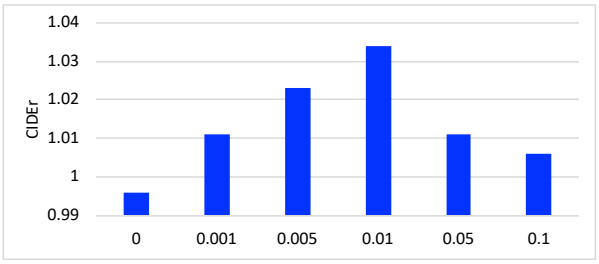

图 4.4: 不同大小的重构权重对于模型性能的影响

4.6 置换顺序的序列化 MNIST 手写数字分类

为了进一步验证我们所提出的自动重构网络能够对循环神经网络起到很好的正则化效果，我们又引入了另一个实验任务——置换顺序的序列化 MNIST 手写数字分类任务。序列化 MNIST 手写数字分类被 Le [65] 等人首先提出，所谓的序列化 MNIST，就是将原先 MNIST 中 28×28 大小的图像拉直变成维度为 784 的一维向量，然后将每个像素逐个输入到循环神经网络中，最后用最后一个时刻（$t = 784$）输出的隐状态 h_{784} 判断这张图像属于哪一个数字。而置换顺序的序列化 MNIST 手写数字分类则是比正常序列化的 MNIST 分类更难的一个任务。在置换顺序的序列化 MNIST 中，原先正常排序的 784 个像素被随机打乱顺序。注意，这里 MNIST 中所有的图像都以相同的置换顺序产生新的序列。比如原先的顺序是 $[1, 2, \cdots, 784]$，现在以 $[784, 783, \cdots, 1]$ 的顺序重新排列像素，产生新的序列化向量。

与图像语义描述的任务相似，我们使用 LSTM 网络对这些逐个输入的像素进行编码，LSTM 中隐状态的参数设置为 128。经过 784 个时刻后，我们使用最后一个时刻 LSTM 网络所输出的隐状态 h_{784} 来对该图像进行分类。我们的自动重构网络也是由另外一个参数设置相同的 LSTM

网络来实现。我们将自动重构网络与另外两个能够对循环神经网络正则化的方法进行比较，一个是 dropout [61] 正则化方法，另一个是 Zoneout [40] 正则化方法。因为前文中所讨论的规定采样 [39]（Scheduled Sampling, SS）是序列解码网络的正则化方法，并不适用于将 LSTM 用于对输入的 MNIST 像素进行编码的情况，所以这里就不与 Scheduled Sampling 方法进行比较。

表4.5展示了多种循环神经网络模型在置换顺序的序列化 MNIST 分类任务上的性能。不加任何正则化方法的 LSTM 网络仅能达到 91.4% 的分类精度，在加入了不同的正则化方法后，分类精度均能有所提升。同时，我们可以发现，在融入了我们的自动重构网络之后，分类的精度达到 93.3%。据我们所知，这是目前在置换顺序的序列化 MNIST 分类任务上最好的结果。这个实验说明了我们提出的自动重构网络结构，对绝大部分使用循环神经网络的任务都适用，而不仅仅适用于图像语义描述中的序列解码网络。

5 工作总结和展望

本文所研究的图像语义描述任务，其本质是从视觉到语言（Visual-to-Language，V2L）的问题。其定义为给定一张图像，我们需要对这张图像的主要内容用一句自然语言进行描述。要求生成的自然描述语句不仅能准备概括图像的主要内容，还要保证所生成的描述语句要通顺流畅，符合语言的语法。图像语义描述作为桥梁，连接了人工智能中的计算机视觉技术与自然语言处理技术，在目前我们国家大力倡导与发展的智慧城市中有着重要的应用，因此该任务有着重要的研究意义、实践意义。本章主要是对本文的主要研究内容、采用的模型以及取得的成果进行总结，重点对本文所提出的自动重构网络进行了回顾。同时，对当前图像语义描述任务所遇到的新问题进行了概述，并对该任务今后的工作重点进行了展望。

5.1 工作总结

本文主要介绍了图像语义描述任务中最常用的编解码网络模型。详细介绍了用于对图像进行特征编码的卷积神经网络，以及用于对图像特征进行解码并生成描述语句的循环神经网络，特别是长短期记忆神经网络（LSTM）。在简述了基础的编解码网络模型的缺陷后，我们引入了视觉注意力机制。随后介绍了视觉注意力机制的原理，及带有注意力机制的编解码网络模型。接着，本文介绍了编解码网络模型中所存在的曝光偏差问题。针对曝光偏差问题，本文提出了一种新的网络模型——自动重构网络（ARNet），这个结构嵌入于编解码网络模型中。具体为，我们用 t 时刻 LSTM 网络输出的隐状态 h_t，通过我们的自动重构网络，去重构出 $t-1$ 时刻的隐状态 h_{t-1}。在任何形式的循环网络中（LSTM、GRU 等），相邻两个时刻的隐状态联系是很紧密的，因为 h_t 是由 h_{t-1} 以及 t 时刻输入的 x_t 经过变换得到的。因此，通过 h_t 来恢复出前一个时刻中隐状态的信息是合乎逻辑的。通过自动网络网络，相邻两个时刻隐状态之间的信息联系更紧密，耦合性也更高。同时，相邻隐状态之间更高层次的语义关系也能够被自动重构网络所挖掘。

我们提出的自动重构网络在权威的 MSCOCO 数据集上，均取得了领先的结果。我们还通过 t-SNE 特征降维技术，以及定量化的实验研究，均显示了自动重构网络能够明显缩小模型训练模式与测试推断模式所产生的隐状态之间的距离，说明这个结构可以有效地缓解曝光偏差问题。

最后，我们还通过置换顺序的序列化 MNIST 手写数字分类实验，验证了自动重构网络结构不仅适用于图像语义描述任务中的编解码网络，还能用于使用循环神经网络对输入序列进行特征编码的编码网络，进一步说明了自动重构网络对循环神经网络有着较好的正则化效果，有着广泛的应用空间。

图 5.1: 目前编解码模型所生成的图像语义描述语句示例图

5.2 工作展望

尽管图像语义描述任务在标准的 MSCOCO 数据集上已经取得了很大的进展，但目前仍有一些问题，需要我们去探索并解决。

第一个问题，目前编解码网络模型所生成的描述语句大多比较简单、单一。如图5.1所示，我们展示了四张内容场景均是浴室的图像，虽然每张图像里具体的内容都不相同，但编解码网络模型所生成的描述语句几乎都一样。对场景的描述不够具体，也不能针对不同的内容生成多样性的描述语句。因此，图像语义描述任务目前朝着生成多样性、差异性的描述语句的方向发展。

第二个问题，目前如 BLEU、CIDEr 这类评价指标已经被提升到很难再提升的地步，我们亟需要一个新的能够评价描述语句多样性的指标，这也是图像语义描述任务的发展方向之一。

第三个问题，目前基于编解码网络的模型需要大量人工标注的描述语句作为标签，以此进行有监督的训练。但在实际生活中，我们并没有那么多标注好的数据，尤其是图像语义描述任务需要更细致的标注，人工标注成本相比较于物体检测任务的标注会更高。例如 MSCOCO 数据集中，有超过十六万张的图像，每张图像都至少有 5 句自然描述语句。像这样大规模人工标注的数据集，其获取成本是非常高昂的。因此，学术界与工业界正探索无监督或者弱监督的图像语义描述方法。

参考文献

[1] Lecun Y., Bottou L., Bengio Y., Haffner P. Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11): 2278-2324.

[2] 杨钊, 陶大鹏, 张树业, 金连文. 大数据下的基于深度神经网的相似汉字识别 [J]. 通信学报, 2014, 35(9): 184-189.

[3] Krizhevsky A., Sutskever I., Hinton G. ImageNet classification with deep convolutional neural networks[C]// International Conference on Neural Information Processing Systems (NIPS), 2012: 1097-1105.

[4] Zoph B., Le Q. Neural architecture search with reinforcement learning[C]// International Conference on Learning Representations (ICLR), 2017.

[5] Mikolov T., Chen K., Corrado G., Dean J. Efficient estimation of word representations in vector space[C]// International Conference on Learning Representations (ICLR), 2013.

[6] Le Q., Mikolov T. Distributed representations of sentences and documents[C]// International Conference on Machine Learning (ICML), 2014: 1188-1196.

[7] 扈中凯. 面向医疗数据的商务智能技术研究 [D]. 浙江大学, 2015.

[8] 国务院印发新一代人工智能发展规划 [J]. 自动化应用, 2017(7): 8.

[9] Cho K., van Merriënboer B., Gülçehre Ç., Bahdanau D., Bougares F., Schwenk H., Bengio Y. Learning phrase representations using rnn encoder-decoder for statistical machine translation[C]// Conference on Empirical Methods in Natural Language Processing (EMNLP), 2014: 1724-1734.

[10] Vinyals O., Toshev A., Bengio S. Show and tell: A neural image caption generator[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015: 3156-3164.

[11] Hochreiter S., Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[12] Schuster M., Paliwal K. Bidirectional recurrent neural network[J]. IEEE Transactions on Signal Processing, 1997, 45(11): 2673-2681.

[13] Chung J., Gulcehre C., Cho K. Empirical evaluation of gated recurrent neural networks on sequence modeling[C]// NIPS Workshop on Deep Learning, 2014.

[14] Bradbury J., Merity S., Xiong C., Socher R. Quasi-Recurrent Neural Networks[C]// International Conference on Learning Representations (ICLR), 2017.

[15] Mao J., Xu W., Yang Y. Deep captioning with multimodal recurrent neural networks(m-rnn)[C]// International Conference on Learning Representations (ICLR), 2015.

[16] Xu K., Ba J., Kiros R., Cho K., Courville A., Salakhutdinov R., Zemel R., Bengio Y. Show, attend and tell: Neural image caption generation with visual attention[C]// International Conference on Machine Learning (ICML), 2015: 2048-2057.

[17] Bahdanau D., Cho K., Bengio Y. Neural machine translation by jointly learning to align and translate[C]// International Conference on Learning Representations (ICLR), 2015.

[18] Lin T., Maire M., Belongie S., Hays J., Perona P., Ramanan D., Dollár P., Zitnick L. Microsoft COCO: Common objects in context[C]// European Conference on Computer Vision (ECCV), 2014: 740-755.

[19] You Q., Jin H., Wang Z., Fang C., Luo J. Image captioning with semantic attention[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 4651-4659.

[20] Lu J., Xiong C., Parikh D., Socher R. Knowing when to look: Adaptive attention via a visual sentinel for image captioning[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 3242-3250.

[21] Girshick R. Fast R-CNN[C]// IEEE International Conference on Computer Vision (ICCV), 2015: 1440-1448.

[22] Ren S., He L., Girshick R., Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017, 39(6): 1137-1149.

[23] 张丽阳, 郝志峰, 肖燕珊, 阮奕邦, 李杰龙. 基于示例加权的稀疏正包多示例学习 [J]. 计算机工程与设计, 2016, 37(5): 1271-1274.

[24] Fang H., Gupta S., Iandola F., Srivastava R., et al. From captions to visual concepts and back[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015: 1473-1482.

[25] 方景龙, 王万良, 王兴起, 龙哲, 祁萌. 求解多示例问题的支持向量数据描述方法 [J]. 电子学报, 2013, 41(4): 763-767.

[26] 丁建浩, 耿卫东, 王毅刚. 基于多部位多示例学习的人体检测 [J]. 模式识别与人工智能, 2012, 25(5): 803-809.

[27] Yao T., Pan Y., Li Y., et al. Boosting image captioning with attributes[C]// IEEE International Conference on Computer Vision (ICCV), 2017: 4904-4912.

[28] Mun J., Cho M., Han B. Text-guided Attention Model for Image Captioning[C]// The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), 2017: 4233-4239.

[29] Papineni K., Roukos S., Ward T., et al. BLEU: A method for automatic evaluation of machine translation[C]// Association for Computational Linguistics (ACL), 2002: 311-318.

[30] Lavie A., Agarwal A. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments[C]// Workshop on Statistical Machine Translation of the Empirical Methods in Natural Language Processing (EMNLP), 2005.

[31] Vedantam R., Zitnick C., Parikh D. CIDEr: Consensus-based image description evaluation[C]// IEEE International Conference on Computer Vision (ICCV), 2015: 4566-4575.

[32] Lin C. ROUGE: A package for automatic evaluation of summaries[C]// Text Summarization Branches Out: Proceedings of the ACL Workshop, 2004: 74-81.

[33] Anderson P., Fernando B., Johnson M., et al. Spice: Semantic propositional image caption evaluation[C]// European Conference on Computer Vision (ECCV), 2016: 382-398.

[34] Ranzato M., Chopra S., Auli M., Zaremba W. Sequence level training with recurrent neural networks[C]// International Conference on Learning Representations (ICLR), 2016.

[35] Williams R. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine Learning, 1992, 8(3): 229-256.

[36] Sutton R., McAllester D., Singh S., et al. Policy gradient methods for reinforcement learning with function approximation[C]// Conference on Neural Information Processing Systems (NIPS), 2000: 1057-1063.

[37] Rennie S., Marcheret E., Mroueh Y., et al. Self-critical sequence training for image captioning[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 1179-1195.

[38] Anderson P., He X., Buehler C., et al. Bottom-Up and Top-Down Attention for Image Captioning and Visual Question Answering[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2018.

[39] Bengio S., Vinyals O., Jaitly N., Shazeer N. Scheduled sampling for sequence prediction with recurrent neural networks[C]// Conference on Neural Information Processing Systems (NIPS), 2015: 1171-1179.

[40] Krueger D., Maharaj T., Kramár J., et al. Zoneout: regularizing rnns by randomly preserving hidden activations[C]// International Conference on Learning Representations (ICLR), 2016.

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

[42] Szegedy C., Liu W., Jia Y., et al. Going deeper with convolutions[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014: 1-9.

[43] Ioffe S., Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]// International Conference on Machine Learning (ICML), 2015: 448-456.

[44] Szegedy C., Vanhoucke V., Ioffe S., Shlens J., Wojna Z. Rethinking the inception architecture for computer vision[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: 2818-2826.

[45] Szegedy C., Ioffe S., Vanhoucke V., Alemi A. Inception-v4, Inception-ResNet and the impact of residual connections on learning[C]// The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI), 2017.

[46] He K., Zhang X., Ren S., Sun J. Deep residual learning for image recognition[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015: 770-778.

[47] Jegou H., Douze M., Schmid C. Hamming embedding and weak geometry consistency for large scale image search-extended version[C]// European Conference on Computer Vision (ECCV), 2008: 304-317.

[48] Huang G., Liu Z., Weinberger K., et al. Densely connected convolutional networks[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017: 2261-2269.

[49] Li X., Lan W., Dong J., Liu H. Adding Chinese Captions to Images[C]// ACM International Conference on Multimedia Retrieval (ICMR), 2016: 271-275.

[50] Yang Z., Yuan Y., Wu Y., Cohen W. W., Salakhutdinov R. Review networks for caption generation[C]// Conference on Neural Information Processing Systems (NIPS), 2016: 2361-2369.

[51] Maron O., Lozano-Pérez T. A framework for multiple instance learning[C]// Conference on Neural Information Processing Systems (NIPS), 1998: 570-576.

[52] Zhang C., Platt C. J., Viola A. P. Multiple instance boosting for object detection[C]// Conference on Neural Information Processing Systems (NIPS), 2005: 1417-1424.

[53] Och J. F. Minimum error rate training in statistical machine translation[C]// Association for Computational Linguistics (ACL), 2003: 160-167.

[54] Jiang W., Ma L., Chen X.,Shen F., Zhang H., Liu W. Learning to Guide Decoding for Image Captioning[C]// The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI), 2018.

[55] Deng J., Dong W., Socher R., Li L., Li K., Li F. ImageNet: A large-scale hierarchical image database[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009: 248-255.

[56] 宋光慧. 基于迁移学习与深度卷积特征的图像标注方法研究 [D]. 浙江大学, 2017.

[57] Cortes C., Vapnik V. Support-Vector Networks[J]. Machine Learning. 1995, 20(3): 273-297.

[58] Hopfield J., Neural networks and physical systems with emergent collective computational abilities[J]. Proceedings of the National Academy of Sciences of the USA, 1982, 79(8): 2554–2558.

[59] Schuster M., Paliwal K. Bidirectional recurrent neural networks. IEEE Transactions Signal Processing, 1997, 45(11): 2673-2681.

[60] Graves A., Mohamed A., Hinton G. Speech recognition with deep recurrent neural networks[C]// IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013: 6645-6649.

[61] Srivastava N., Hinton G., Krizhevsky A., Sutskever I., Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting[J]. Journal of Machine Learning Research, 2014, 15(6): 1929-1958.

[62] Maaten L., Hinton G. Visualizing data using t-SNE[J]. Journal of Machine Learning Research. 2008, 9(11): 2579-2605.

[63] Andrej K., Li F. Deep Visual-Semantic Alignments for Generating Image Descriptions[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2017, 39(4): 664-676.

[64] Kingma D., Ba J. Adam: A Method for Stochastic Optimization[C]// International Conference on Learning Representations (ICLR), 2015.

[65] Le Q. V., Jaitly N., Hinton G. E. A simple way to initialize recurrent networks of rectified linear units. ArXiv preprint arXiv:1504.00941, 2015.

[66] Maas A., Hannun A., Ng A. Rectifier nonlinearities improve neural network acoustic models[C]// ICML Workshop on Deep Learning for Audio, Speech and Language Processing, 2014.

攻读硕士期间科研经历与研究成果

1. **Xinpeng Chen**, Jingyuan Chen, Lin Ma, Jian Yao, Wei Liu, Jiebo Luo, Tong Zhang. Fine-grained Video Attractiveness Prediction Using Multimodal Deep Learning on a Large Real-world Dataset. In WWW'18 campanion: The 2018 Web Conference Campanion, Lyon, France, April 23-27, 2018.
2. **Xinpeng Chen**, Lin Ma, Wenhao Jiang, Jian Yao, Wei Liu. Regularizing RNNs for Caption Generation by Reconstructing The Past with The Present. IEEE Conference on Computer Vision and Pattern Recongition (CVPR'18), Salt Lake City, USA, 2018.

致　　谢

时光飞逝，转眼已三年，我的硕士生涯已接近尾声。如果要用一个词来概括这三年的历程，那就是成长。借此课题完结之际，感谢成长路上陪我一起走过的每位敬爱的老师和亲爱的同学。这三年中，我们共同为青春奋斗，我们共同在黑夜中祈求黎明，我们共同在成功时把酒言欢，我们共同在陪伴中相知相交。无论未来我们携手同行还是相忘于江湖，我都深深的祝福你们。我将永远珍藏我们之间的点滴回忆，这是我一生的财富。

感谢我的学术领路人——姚剑教授。三年前，是姚老师带领我走进计算机视觉的大门。姚老师不仅学术上深有造诣，而且为人谦和、善解人意。在我刚开始入门的时候，是姚老师细心的指导我阅读文献的方法与技巧；在我研究中遇到困难的时候，是姚老师鼓励我勇敢的前进。感谢姚老师对我的耐心与包容，让我更坚定的在学术道路上继续前进。

感谢 CVRS 实验室的小伙伴们，和你们在一起的日子就像一家人，这三年里与你们一起充满了快乐的回忆。感谢张考师兄在我刚进组的时候对我的照顾与指导。感谢 213 机房的朱吉、张瑞倩、董颖青、涂金戈，是你们带领我融入这个大家庭，你们的陪伴与鼓励是我战胜困难的动力。感谢刘亚辉、刘媛，研一暑假我们组队参加智慧城市比赛，一起奋斗的日子终身难忘。感谢已经毕业的廖岩岩、黄中帅，与你俩一起过生日的时候开心又幸福。

感谢日立制作所的童彬老师给我前往东京实习的机会，在带领我体验日本文化的同时，手把手的教我基础知识，使我在多模态这个研究方向打下了坚实的基础。感谢腾讯 AI Lab 的马林老师、姜文浩老师、刘威老师，非常幸运能够进入 AI Lab 学习，在你们细心的指导下，我能够快速的成长并能在学术上有所收获。感谢新加坡国立大学的陈静远学姐对我的帮助，一起讨论一起赶论文的时光，虽然辛苦，但快乐而充实。

感谢我多年的朋友石星辰和杭天宇，岁月如歌，友情如酒，希望我们的友谊天长地久。感谢我的父母对我的关爱，无论是在精神上还是经济上，他们始终是我最坚强的后盾。三年过去了，默默辛劳的他们又增添了几丝白发。学生生涯的终结，它不仅仅是一个结束，更也是责任的开始。愿亲爱的父母身体健康，万事顺心。

感谢百忙之中参与审阅、评议本论文各位老师，也向参与本人论文答辩的各位老师表示由衷的感谢。

最后，感谢武汉大学、遥感学院给我们营造提供的学习与成长的平台。珞珈山水一程，晚辈三生有幸，我们江湖再见！

武汉大学学位论文使用授权协议书

本学位论文作者愿意遵守武汉大学关于保存、使用学位论文的管理办法及规定, 即：学校有权保存学位论文的印刷本和电子版, 并提供文献检索与阅览服务；学校可以采用影印、缩印、数字化或其它复制手段保存论文；在以教学与科研服务为目的前提下, 学校可以在校园网内公布部分及全部内容.

1、 在本论文提交当年, 同意在校园网内以及中国高等教育文献保障系统 (CALIS) 高校学位论文系统提供查询及前十六页浏览服务.

2、 在本论文提交 □ 当年/ □ 一年/ □ 两年 / □ 三年/ □ 五年以后, 同意在校园网内允许读者在线浏览并下载全文, 学校可以为存在馆际合作关系的兄弟高校用户提供文献传递服务和交换服务.(保密论文解密后遵守此规定)

论文作者 (签名)：_____

学　　号：_____

学　　院：_____

日期：　　年　　月　　日