基于区域注意力机制的图像描述算法

摘要

图像描述是计算机视觉与自然语言处理的重要交叉领域，是通向机器智能的许多日常情景应用，如图像检索，儿童教育和视力受损人士的生活辅助等的至关重要的一步。随着计算机的硬件性能大幅提升，海量存储空间和 GPU 超强的运算能力，深度神经网络快速崛起，并在图片分类、目标检测、目标分割等领域取得了巨大突破，为图像描述算法的研究奠定了基础。本文使用深度神经网络对图片描述任务进行设计和建模，并实现了 web 服务器端程序，可为图片自动生成对应的文字描述。

本文提出了基于区域注意力的图像描述算法。首先使用 Faster R-CNN 模型中的 RPN 生成候选区域，然后通过在最后一层共享卷积层进行 roi pooling 提取图片特征，最后将图像特征通过注意力机制提供给 LSTM 生成描述语句。通过将描述生成过程进行可视化，显示了在描述生成过程中注意力的变化，实验表明算法在多个指标下达到目前技术前沿水平。

在此基础上基于 Flask 搭建了图像描述生成的服务器端程序，用户可通过浏览器上传图片，查看机器描述并给出评分。管理员可通过后台查看用户上传图片，对机器描述给出的评分，及描述生成过程中的注意力分布，便于查看算法性能，对算法做进一步改进。

关键词：图像描述；卷积神经网络；LSTM；注意力机制

1．项目背景与意义

图像描述，也称为图像语义理解，是计算机视觉与自然语言处理的重要交叉领域，是通向机器智能的许多日常情景应用，如图像检索，儿童教育和视力受损人士的生活辅助等的至关重要的一步[1]。2016年4月，微软上线了人工智能应用Seeing AI，通过对摄像头采集的图像进行分析，并将分析结果以语音的方式传达给用户，使用户可以“听到”周围环境情况，从而为视力缺陷者带来帮助。

相较于传统的场景识别、目标检测、目标识别算法，图像语义理解不仅关注图片中的物体，而且给出了物体的属性及物体间的关系，其目标是根据图片内容总结出语义一致的可理解的句子。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，而在图像描述任务中，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。除此之外，图片中含有复杂的场景信息，一张图片往往可以从多个不同角度使用不同词语来描述，这意味着与分类任务中可以直接判断模型输出正确与否不同，判断图像描述模型生成的描述语句是否正确是比较困难的。

近年来，手机的普及使得音视频的采集变得轻而易举，而互联网基础设施趋于完善则极大地便利了数据的传播与收集。与此同时，GPU在图像处理上表现出的强大运算能力，使得深度神经网络的训练成为可能。海量的训练样本与充足的计算资源使得深度学习技术得以蓬勃发展，人们在这个领域获得了瞩目的成就。2016年3月AlphaGo挑战李世石成功使得深度学习变得炙手可热，而早在2012年，AlexNet获得ILSVRC比赛冠军，标志着深度学习在计算机视觉领域取得了突破性进展，深度学习就已开始在学术界受到广泛关注。与此同时，自然语言处理同样作为深度学习重点研究和应用的领域，近年来也取得了许多有影响力的成果，语音交互正走进人们的日常。值此时代背景下，图像描述任务逐渐吸引了研究者的注意。

图像描述算法的研究早在十多年前就已在计算机视觉与自然语言处理领域开展，主要可分为三类。基于模板的算法[3-10]首先假定了语法模型，然后根据图片中检测到的物体及其属性填空，但是这种方法限制了输出形式，生成的语句也并不总是可理解的。基于检索的算法[11-15]选择训练集中与测试图片最为相似的图片的描述作为结果输出，无法生成新的描述语句。实现上首先提取图像特征和语句特征，然后优化将图像特征和语句特征映射到语义特征空间的编码模型，通过计算图片与描述之间的距离来从训练语料中选择图片描述。随着当下深度神经网络在机器翻译任务上取得成功，基于编解码器的算法[16-21]被提出，可以生成未在训练集上出现的结构更为灵活的句子。在编解码器框架下，原始图片首先经过卷积神经网络提取语义特征，通常取倒数第二个全连接层的输出作为

全局视觉特征向量。然后使用循环神经网络根据视觉特征向量生成对应描述，通常会使用相较于RNN更易于训练且性能更好的LSTM或GRU。

m-RNN[16]开创性地将卷积神经网络和循环神经网络结合起来，以解决图像描述和图像检索等问题。首先使用CNN对输入图像进行编码，RNN对已生成语句进行编码，然后将图像特征和语句特征送入混合模型预测下一单词。网络结构如图2.1所示。

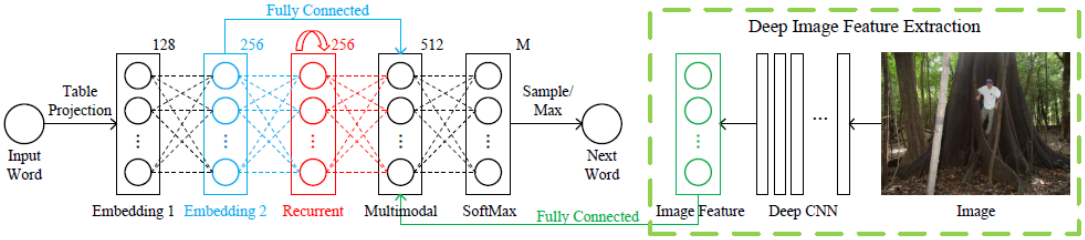


图 1.1 m-RNN 结构图

所使用的RNN与传统RNN为隐层状态与输入各加权重生成当前时刻的隐层状态不同，激活函数选择了ReLU来应对RNN训练中梯度消失的问题，隐层状态的更新公式为

$$h(t)=\mathrm{Re}LU(W_h\bullet h(t-1)+x(t)) \tag{1.1}$$

混合模型（multimodal layer）采用了隐层单元为 512 的 RNN，其输入为

$$m(t)=g(V_x\bullet x(t)+V_h\bullet h(t)+V_I\bullet I) \tag{1.2}$$

$$g(x)=1.7159\tanh(\frac{2}{3}x) \tag{1.3}$$

其中 I 是从 AlexNet 第 7 层提取的图像特征。$g(x)$ 相较于传统的 $\tanh$ 激活函数，在反向传播的时候梯度更多地落在非饱和区，因而训练速度更快。m-RNN 的损失函数基于 PPL，定义为

$$Loss=-\frac{1}{N}\sum_{i=1}^{N}\sum_{n=1}^{L}\log_2 P(w_n^i\mid w_{1:n-1}^i,I^i)+\|\theta\|_2^2 \tag{1.4}$$

$\|\theta\|_2^2$ 是正则项。训练目标是最小化损失函数，即最大化生成图像的对应描述的概率。

NIC[17]采用编解码器（Encode-Decode）结构，首先使用 CNN 将输入图片编码成一个固定长度的向量表示，然后使用 RNN 来将文字描述解码出来。RNN 首先使用 CNN 提取的固定长度的特征生成初始状态向量，然后根据当前的输入单词预测下一输出单词，再将输出作为输入预测下一输出。相较于 m-RNN 模型，NIC 模型选择了性能更好的 LSTM 来代替 RNN，图像特征提取部分使用了比 AlexNet 的网络结构更有效的 inception_v3，将机器翻译中的编解码器结构应用到图像描述任务上，CNN 提取的图像特征数据只在开始输入一次。网络结构如图 1.2 所示。

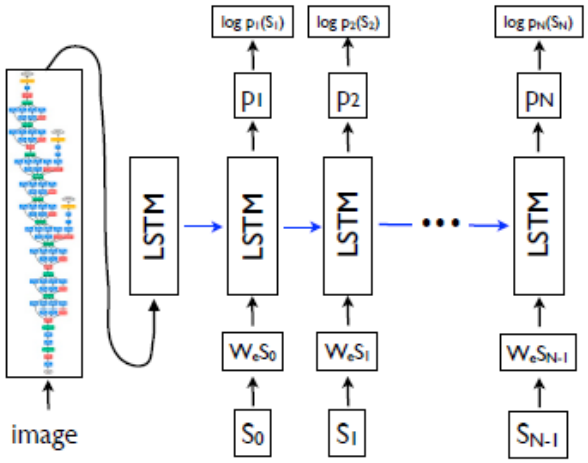


图 1.2 NIC 结构图

NeuralTalk[18]首先使用 R-CNN 对图片中的物体进行检测，选取最可能出现的前 19 个物体的所在区域及原图共 20 个区域，送入 CNN 中编码成长度为 4096 的向量，然后通过混合层将每张图片编码成由 h 维向量构成的集合。于此同时对描述语句进行词向量编码后送入 BRNN（Bidirectional Recurrent Neural Network）提取每个单词对应的 h 维语义向量。最后设计损失函数将图片和短语对应起来。效果如图所示。然而在测试阶段，与 NIC 类似，使用 CNN 从整张图片中提取特征，然后送入 RNN 生成描述语句。

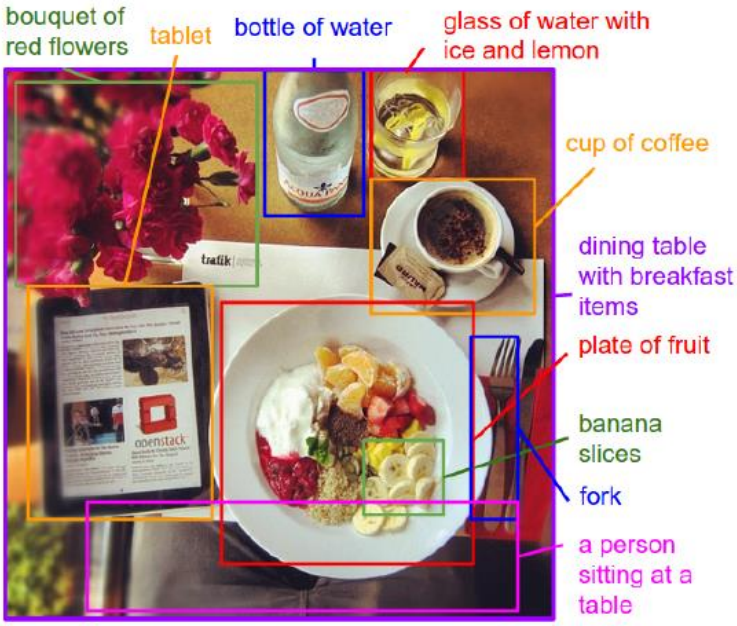


图 1.3 NeuralTalk 效果图

ATT-NIC[19]针对 NIC 模型中存在编码向量长度固定，对于短描述足够使用的编码长度，对于长描述可能难以保留全部的必要信息的问题，提出使用注意力机制，每次只注意图片中的一部分，大幅提升了模型性能。算法在使用卷积神经网络提取图片特征时，不再只是提取全局视觉特征向量，而是为图中的每个子区域都生成一个特征向量，构成一个特征向量集。图像特征也不再源于倒数第二个

全连接层，而是更前面包含了空间信息的卷积层。然后在描述生成阶段，注意力模型会根据循环神经网络的隐层状态生成当前上下文，即区域特征向量的加权平均，并提供给循环神经网络生成下一个词。

不同于上述使用图像特征生成描述语句的模型，另一类模型[20,21]首先检测图中有哪些物体，然后将这些词组合成句子。其中[21]在CNN-RNN的框架下引入了高层语义特征,指出直接使用CNN提取的图像特征生成描述语句的成功不是因为避免了高层语义信息的表达，将图像特征转换成语义概念输入RNN能够带来模型性能的提升。网络结构如图1.4所示。

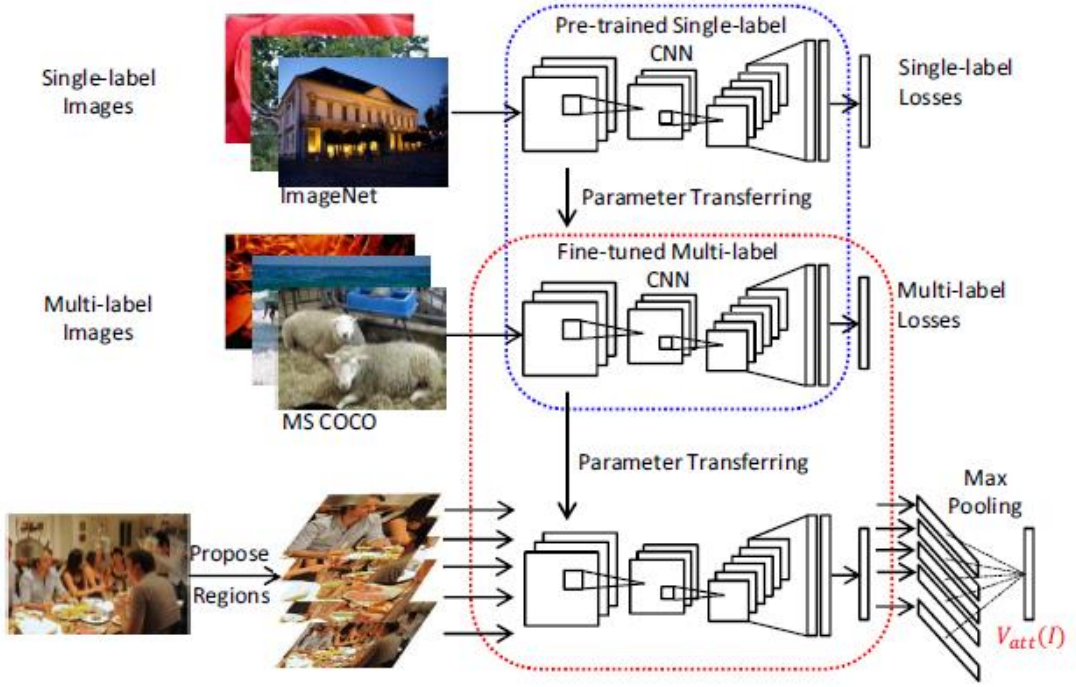


图 1.4 使用高层语义特征生成图像描述

相较于 NIC，模型使用了多标签的 CNN 提取图像的高层语义特征,如"狗","奔跑"等名词,动词或者形容词在图中出现的概率，作为 RNN 的输入。通过对单复数形式和动词时态不加区分，即将 ride 和 riding，bag 和 bags 均映射到词根，将词汇表大小压缩到 256。实际训练中，首先用在 ImageNet 上预训练的 VGGNet 作为初始模型，然后根据最后一个全连接层的输出，使用 softmax 分类器给出在 256 个标签上的分类结果。使用多标签数据进行优化训练，损失函数定义为

$$Loss=\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{c}\log(1+e^{(-y_{ij}p_{ij})}) \tag{1.5}$$

其中 $y_i=\{y_{i1},y_{i2},...,y_{ic}\}$，$y_{ij}\in\{0,1\}$ 代表第 i 张图片的标签，$p_i=\{p_{i1},p_{i2},...,p_{ic}\}$ 是 c 个 softmax 层输出的具有各标签的概率。

在为图片生成描述时，为了预测图中每个区域的标签，首先使用 MCG[22]算法将图像分成许多区域，然后使用归一化剪枝算法将这些区域归到 10 个簇中，然后从每个簇中选择 5 个最可能的区域，连同原图共 51 个区域，送入 CNN 中进

行特征提取，最后使用最大池化生成图片的高层语义特征向量$V_{\text{att}}(I)$。

与此同时，目标检测作为图像描述的相关领域，近年来也取得了较大进展。R-CNN[23]在 2013 年由 Ross Girshick 提出，在 ILSVRC2013 上 mAP=31.4%，在 VOC 2012 上 mAP=53.3%。相较于 Overfeat 使用滑动窗口法+CNN 来完成目标检测，在 ILSVRC2013 上 mAP=24.3%；UVA 使用 selective search+SIFT+SVM 来完成目标检测，在 VOC 2012 上 mAP=35.1%，检测精度有了显著提升。算法首先通过 selective search[24]对图像进行分割，生成相互重合的多个候选区域，然后将每个候选区域缩放到 227*227 大小，送入 CNN 中提取维度为 4096 的特征向量，最后使用 SVM 进行分类。通过使用非最大抑制，即如果有个区域跟它的 IoU 大于某一阈值，但是通过 SVM 给的分更高，丢掉它；以及外接矩形回归，即在 pool5 后接全连接层拟合真正的（x,y,w,h），给出最终结果。Selective search 首先将图片划分为许多颜色变化剧烈、缓慢和基本不变的区域，然后将这些区域依照相似度组织成一棵树。其中相似度由颜色相似度、纹理相似度、大小相似度和位置相似度加权得到。最后选择被频繁定位的区域作为候选区域。因为依照颜色认为这里有东西，依照纹理认为这里也有东西，那么这里很有可能真的有东西。

Fast R-CNN[25]在2015年4月由Ross Girshick提出，在VOC2012上mAP=66%(vs. 62% for R-CNN)。通过在同一张图片的不同候选区域之间共享特征信息，不包括使用selective search 生成候选区域的时间，将检测速度由R-CNN的47s/image提升到0.32s/image。相较于R-CNN，主要提出了RoI pooling。首先让图片经过CNN获取特征图，然后将原图上的候选区域按相对位置直接映射到特征图上，将特征图上的对应区域划分为7*7的网格，在每一个网格内做max pooling，即为RoI pooling。RoI pooling为每个候选区域生成了固定大小的特征。之后通过多个全连接层和softmax分类器得到的分类结果，并通过线性回归得到每一个类对应的外接矩形框。

Faster R-CNN[26]在2015年6月由任少卿等提出，在VOC2007上mAP=59.9%（vs. 58.7% for Fast R-CNN）。提出在候选区域生成上使用RPN(Region Proposal Network)来代替selective search，然后使用fast R-CNN来检测。通过使RPN和fast R-CNN共享卷积层，提出了一个针对目标检测的统一网络，将候选区域生成的时间从2s(CPU实现)缩减到10ms，将候选区域的数量从2000降到了300，将全部步骤的处理速度提升到5fps。为了使RPN能够准确检测出不同大小和长宽比的候选区域，提出了“锚点”的概念。对共享卷积层的最后一层特征图上每一个3*3的区域，预测中心在滑动窗口的中心，尺度取[0.5, 1.0, 2.0]，长宽比取[0.5, 1.0, 2.0]，共k=9个锚点上，区域建议的类别（2*9=18维）及位置（4*9=36维）。在一张W*H (typically ~2400)的特征图上，有WHk 个anchors。

本文在对图像描述算法的发展现状进行综述的基础上，提出基于区域注意力机制的图像描述算法，对注意力机制在语句生成过程中的作用进行了探索。并设计了 web 应用程序，提供可视化界面，便于发现算法存在的问题，进一步完善算法。

2. 深度神经网络

2.1 概述

深度学习的起源可追溯到 1958 年，Frank 基于生物神经科学提出了感知机模型，从此开启了人工神经网络的研究。1969 年，M.Minsky 和 S.Papert 将感知机模型用于线性分类，但受制于当时的训练手段和硬件运算能力，这一发现并未引起大家大多关注。1962 年 Hubei 和 Wiesel 通过对猫视觉皮层细胞的研究，提出了感受野的概念。1986 年 Rumel Hart、Hinton 和 Williams 提出了反向传播算法，激起了神经网络研究的巨大热情，但 BP 神经网络容易陷入局部极小值或者出现过拟合现象，特别是在增加网络层数时，而同时期以支持向量机为代表的传统机器学习算法也有了突破性的进展，再加上数据量无法满足训练深层神经网络的需求，使得人工神经网络再次进入寒冬。1998 年 LeCun 等人提出 LeNet 模型用于手写字母识别，开启了卷积神经网络的时代。2006 年 Hinton 等人提出了深度置信网络，并发现可以通过“逐层初始化”的方式进行训练，不再被训练方法所局限。正是由于训练方法的改善，发现含有多层隐藏层的人工神经网络的学习能力很强，特别是对数据的特征的提取能力。继 2006 年以后，深度学习的研究在学术界受到持续欢迎。

2.2 卷积神经网络

卷积神经网络是深度学习的重要模型之一，由卷积层、池化层、全连接层构成。该模型通过局域感受野、权值共享和次抽样处理，从而具有位移、缩放和扭曲不变性。局域感受野可以提取初级的视觉特征，权值共享可以减少神经元之间的参数，减少训练学习的时间，次抽样减小了特征分辨率，实现了扭曲不变性。卷积神经网络通过训练数据自主隐式地学习数据中的特征，避免了人为选取特征不理想的缺陷，在生物、医疗、交通等领域有着广泛应用。

2.2.1 激活函数

激活函数为神经网络提供了非线性建模能力，使得模型几乎能够拟合任意函

数映射，是神经网络能解决非线性问题关键。图像的卷积或全连接运算的本质是为每个输入值赋予一个权重，仅能表达线性映射，即便增加网络深度也难以有效对实际环境中非线性分布的数据进行建模，因此激活函数是神经网络中不可或缺的组成部分。常用激活函数包括 Sigmoid 函数、Tanh 函数、ReLU 函数、PReLU 激活函数等。

Sigmoid 函数是单位阶跃函数的平滑性表示，将输入压缩到 0 和 1 之间，输入很大时输出结果接近于 1。在输入接近负无穷的时候输出结果接近于零。被定义为

$$\sigma(x)=\frac{1}{1+e^{-x}} \tag{2.1}$$

其梯度为

$$\sigma'(x)=\sigma(x)(1-\sigma(x)) \tag{2.2}$$

在梯度反向传播的过程中，激活函数的导数最大值为 0.25，当 x 远离原点时，激活函数的导数数值趋近于零，导致误差无法回传，即出现梯度消失问题，最终得到的网络前几层只是随机变换，只在最后几层才真正在做分类。另一方面，Sigmoid 函数的输出大于零，本层输出通过权重关联到下一层，反向传播时，与下一层神经元关联的权重得到的权值更新要么同时为正，要么同时为负，如果这些权重有的需要增加有的需要减小来使损失函数下降，那么就会出现 zig zag 现象，导致梯度更新时发生震荡。

tanh 激活函数与 Sigmoid 函数类似，但是输出为（-1,1）区间，其定义为

$$\tanh(x)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}=2*\text{sigmoid}(2x)\text{-}1 \tag{2.3}$$

尽管存在输入趋近于负无穷或者正无穷时，函数导数数值接近于 0 而带来的梯度消失问题，但是它没有 zigzag 现象，收敛更快。

ReLU[30]是修正线性单元的简称，其定义为

$$\text{Re}LU(x)=\begin{cases}x & x>0\\ 0 & x\le 0\end{cases} \tag{2.4}$$

当 x<0 时，导数恒为 0，导致神经元不再更新，即出现神经元死亡，但也因此可以实现参数稀疏。而当 x>0 时，导数恒为 1，有效改善了梯度消失的问题，但同样存在 zigzag 现象。与 Sigmoid 函数和 Tanh 函数相比，使用 ReLU 作为激活函数训练起来会快很多。例如将深度神经网络训练至 25%的误差率，Re LU 激活函数只需 5 轮迭代，而使用 tanh 作为激活函数则需要 35 次迭代。

PReLU[31]为解决使用 ReLU 激活函数带来的可能存在的神经元死亡的问题而设计，被定义为

$$P\text{Re}LU(x)=\begin{cases}x & x>0\\ ax & x\le 0\end{cases} \tag{2.5}$$

其中a参与梯度下降的优化过程。相较于 ReLU 改善了 zigzag 问题，在许多场景中展现出了更好的效果。

ELU[32]相较于 PReLU 性能更好，但是计算代价稍高，被定义为

$$elu(x)=\begin{cases} x & x>0 \\ a(e^{x}-1) & x\le 0 \end{cases} \tag{2.6}$$

SELU [33]具有自归一化作用，即可自动将输出归一化到均值为 0，方差为 1，能够加速收敛，最终效果与采用 batch normal 相近。定义为

$$selu(x)=scale*\begin{cases} x & x>0 \\ \alpha(e^{x}-1) & x\le 0 \end{cases} \tag{2.7}$$

其中

$$\begin{aligned} scale &= 1.0507009873554804934193349852946 \\ \alpha &= 1.6732632423543772848170429916717 \end{aligned} \tag{2.8}$$

Swish[34]由 Google 在 2017 年 10 月提出，函数曲线与 ReLU 相近，拥有不饱和,光滑,非单调性的特征,在不同的数据集上都表现出了优于当前激活函数的性能。被定义为

$$swish(x)=x*sigmoid(x) \tag{2.9}$$

2.2.2 权重初始化

不同的权重初始化方式影响着网络的收敛速度甚至是收敛性。常用的初始化方式包括常量初始化、高斯分布初始化、均匀分布初始化、xavier 初始化[35]和 msra 初始化[31]常量初始化具有强再现性，在样本相同的两次训练中具有相同的结果，在早期训练神经网络中采用，其中全零初始化大部分情况下会导致学习失败，得不到可用的模型。高斯分布初始化相较于固定值初始化，通过引入随机性增强了网络的容纳能力，有效提升了网络收敛性；xavier 初始化解决了在实际应用中，使用高斯分布初始化时，标准差如何选择的问题，使得训练速度和收敛性有了明显提升。w 符合动态方差的独立高斯分布，即

$$w\sim G(0,\sqrt{1/n}^{2}) \tag{2.10}$$

其中 n 是本层神经元的数量。msra 初始化相较于 xavier 性能提升了一个数量级，定义为

$$w\sim G(0,\sqrt{2/n}^{2}) \tag{2.11}$$

2.2.3 正则、dropout 与 batch normalization

在训练网络的过程中，随着迭代次数的增多，可能出现模型对训练集拟合的很好，但在测试集上表现很差的情况。这种现象是因为模型过于精细复杂，规则

过于严格，以至于测试数据稍微与同类的训练数据有所差别，就认为不属于这个类别。采用 L1 和 L2 正则化方法，修改损失函数，具体通过权重衰减实现，可缓解这一问题。Dropout 同样为防止过拟合而提出。具体实现方式是在网络训练过程中，以概率 p 随机将某一个节点的输出清零，在反向传播时也不对该节点进行权值更新。通过使每个神经元的输入值都不再依赖固定的神经元，迫使网络不断地学习随机神经元子集中的有用链接，来增强网络的泛化能力。

批归一化（batch normalization）[36]针对图像数据的每一维一般都是 0-255 之间的数字，而神经网络模型在初始化的时候，权重 W 是随机采样生成的，在使用梯度下降时，可能需要很多次迭代才能找到最佳分割，从而带来的求解速率慢的问题而设计。通过将输入的每一维度减去均值，除以标准差，使数据点不再只分布在第一象限，增大随机分界面落入数据分布的概率来加速收敛。由于均值和方差是在当前迭代的 batch 中计算的，因此算法被命名为 batch normalization。当每层的输入数据的尺度不一致时，对应权重需要的学习率是不一样的，通常需要使用最小的学习率来保证损失函数稳定下降。批归一化将输入数据尺度保持一致，使得使用较高的学习率进行优化成为可能。除此之外，由于导致过拟合的位置往往在数据边界处，如果初始化权重就已经落在数据内部，过拟合现象就可以得到一定的缓解，因此可以移除或使用较低的 dropout,降低 L2 权重衰减系数。

2.2.4 优化算法

梯度下降是神经网络优化中最常用的算法[37]。传统梯度下降算法[38]难以确定合适的学习率，太小的学习率会导致网络收敛过于缓慢，而太大又会导致损失函数出现震荡，甚至梯度发散。此外，在神经网络中，最小化非凸损失函数的另一个关键挑战是脱离鞍点，这些鞍点的梯度在所有维度上接近于零。传统梯度下降算法存在收敛到局部最优及难以脱离鞍点的问题，其梯度更新公式为

$$w_t = w_t - \eta \frac{1}{m} \sum_{i=1}^{m} \Delta w_i \tag{2.12}$$

其中 m 是样本数量，η 是学习率。在处理大型数据集时，由于进行一次更新需要计算整个数据集梯度，因此速度很慢且难以控制，甚至导致内存溢出。随机梯度下降每次只取一个样本用于计算梯度并进行权重更新，执行速度更快，有助于发现更优的局部最小值，但是频繁更新使得参数具有高方差，损失函数出现大幅波动，难以获得最小值。实际应用中常使用小批量梯度下降（Mini Batch Gradient Descent），对一个批次中的 n 个样本执行一次更新。通常一个批次的大小为 32 到 256，以减少参数更新的波动，最终得到更稳定的收敛结果。这种

方法有时也被称为 SGD。

Momentum SGD[39]针对 SGD 方法中参数的高方差振荡使得网络难以稳定收敛的问题，通过优化相关方向的训练并弱化无关方向的振荡，从而得到更快且稳定的收敛。并且当损失下降到局部最低点时，在动量作用下可能会越过局部最优落到更优的局部最优。权重更新公式为

$$\begin{aligned} m_t &= \mu * m_{t-1} + \eta \Delta w_t \\ w_t &= w_{t-1} - m_t \end{aligned} \tag{2.13}$$

其中 μ 是动量项，通常设为 0.9 或某个相近值，η 是学习率。

Nesterov 梯度加速法[40]针对动量方法在损失到达最低点时，由于高动量导致错过最小值的问题，通过先根据之前的动量预测下一位置，然后根据下一位置的梯度更新动量。

$$\begin{aligned} m_t &= \mu * m_{t-1} + \eta \nabla(w_t) J(w_{t-1} - \mu * m_{t-1}) \\ w_t &= w_{t-1} - m_t \end{aligned} \tag{2.14}$$

RMSprop 是 Geoff Hinton 提出的未公开发表的算法[37]。针对相同的学习率并不适用于所有的参数更新，不常被更新的权重应使用更大的学习率以跳出当前的局部极小的情况设计，有效解决了陷入鞍点的问题。提出梯度更新公式

$$\begin{aligned} E[(\Delta w)^2]_t &= 0.9E[(\Delta w)^2]_{t-1} + 0.1(\Delta w)_t^2 \\ w_t &= w_{t-1} - \frac{\eta}{\sqrt{E[(\Delta w)^2]_t + \varepsilon}} \odot \Delta w_t \end{aligned} \tag{2.15}$$

其中 $\eta = 1e^{-3}$，算法基本不再依赖学习率。

Adam[41]算法与其他自适应学习率算法相比，收敛速度更快，并且可以纠正其它优化技术中存在的梯度消失、收敛过慢或是高方差的参数更新导致损失函数波动较大等问题。算法使用了平方梯度的指数衰减平均值来描述各权重的更新频率，梯度的指数衰减平均值来描述动量，梯度更新公式为

$$\begin{aligned} m_t &= \frac{\beta_1 m_{t-1} + (1-\beta_1)\Delta w_t}{1-\beta_1^t} \\ v_t &= \frac{\beta_2 v_{t-1} + (1-\beta_2)\Delta w_t^2}{1-\beta_2^t} \\ w_t &= w_{t-1} - \frac{\eta}{\sqrt{v_t} + \varepsilon} m_t \end{aligned} \tag{2.16}$$

其中 β_1=0.9， β_2=0.999， ε=$1e^{-8}$。

2.2.5 经典网络

LeNet-5[42]作为卷积神经网络的开端在 1998 年由 LeCun 提出，它是一个由卷积层、池化层、全连接层构成的 5 层神经网络，如图所示。首先对 28*28 的

mnist 手写数字图片做宽度为 2 的 padding 获得 32*32 的输入，通过使用 6 个 5*5 的卷积核进行卷积获得 6*28*28 的特征图，经过 2*2 的池化将特征图维度降为 6*14*14，通过使用 16 个 5*5 的卷积核进行卷积获得 16*10*10 的特征图，再经过 2*2 的池化将特征图维度降为 16*5*5，通过使用 120 个 5*5 的卷积核进行卷积获得 120 维特征向量，然后通过全连接获得维度为 84 的特征输出，最后使用欧式径向基函数获得维度为 10 的输出，即对应 0~9 的概率。受当时计算机计算能力的限制，从 S2 到 C3 没有对特征图的全部 channel 进行卷积，而是使用了复杂的局部连接来减小计算量。损失函数没有使用 softmax 和交叉熵，而是使用了欧式径向基函数和均方误差。

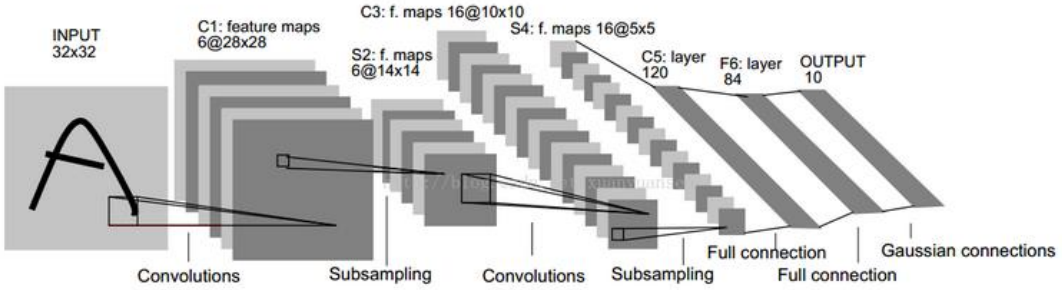


图 2.1 LeNet-5 结构图

2012 年 Alexnet[43]被提出，碾压性地获得了 2012 年 ILSVRC[44]大赛冠军。事实上其最大优势在于使用了 NVIDIA 的 GPU，大大缩短了训练所需时间，使得训练大型神经网络成为可能。提出了有效抑制过拟合的方法 dropout。网络采用了较大的深度，由于 GPU 显存的限制，将网络从 depth 维度拆成了两份，通过两块 GPU 并行进行训练，最后将第 2 块 GPU 上得到的 2048 维特征向量拼回第 1 块 GPU，获得最终的 1000 维分类结果。网络结构如图所示。

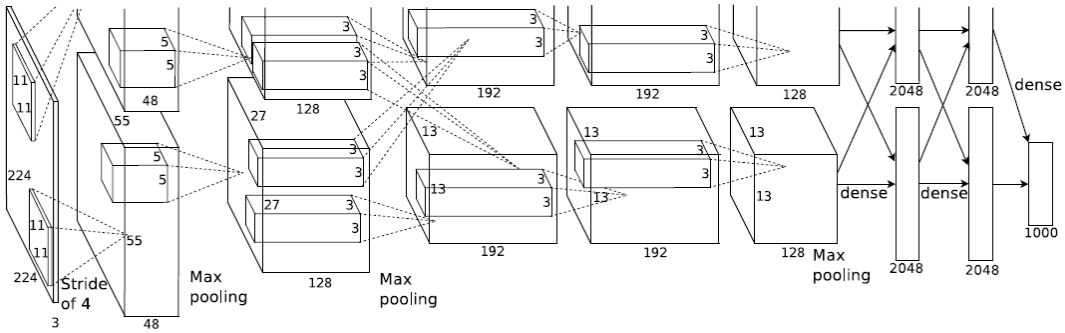


图 2.2 AlexNet 结构图

2014 年 VGG[45]被提出，通过将网络深度大幅提高到 19 层，表现出了比 AlexNet 更好的性能，取得了 ILSVRC2014 比赛分类项目的第二名和定位项目的第一名。其结构简洁，整个网络都使用 3*3 的卷积核和 2*2 的最大池化，因而在提出之后广泛应用于目标检测、目标分割、风格迁移等需要提取图片特征以及神经网络可视化等场合。

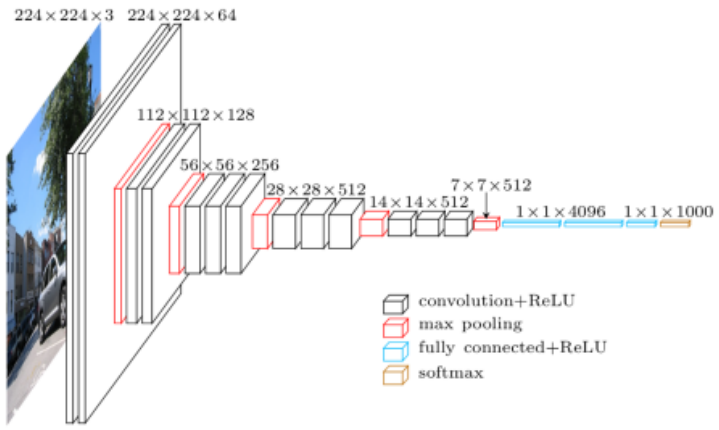


图 2.3 VGG 结构图

GoogleNet[46]是 ILSVRC2014 比赛冠军模型，它证明用更多的卷积，更深的层次可以得到更好的性能。网络深度达到了 22 层，采用了 Inception 结构，将 1x1，3x3，5x5 的卷积和 3x3 的池化堆叠在一起，提升了网络的宽度，增加了网络对尺度的适应性。实际应用中，在 3x3，5x5 的卷积之前，3x3 的池化之后使用了 1*1 的卷积将特征图在 channel 维度上进行压缩，如图所示。相较于使用了 6 千万个权重的 AlexNet，GoogleNet 权重数量仅为 5 百万，减少了 12 倍，而分类性能有了显著提升；VGG 的权重数量是 AlexNet 的 3 倍，而 GoogleNet 与其性能相近,略有胜出。此后 Google 对 Inception 模型进行改进,在 Inceptionv2[36]提出使用 batch normalization 来加速计算，并同 VGG 一样使用 2 个 3*3 的卷积替代 inception 模块中的 5x5 卷积以降低参数数量；Inceptionv3[47]提出非对称的卷积结构，进一步将 3*3 的卷积分解成两个 1*3，3*1 的卷积，使得权重数量进一步减少；Inceptionv4[48]将 Inception 模块进一步优化，并将其与引入了残差结构的 inception-resnet-v2 对比，发现 ResNet 的结构可以极大地加速训练，同时性能也有提升，但在不引入残差结构的基础上也能达到和 inception-resnet-v2 结构相似的结果，说明“要想得到深度卷积网络必须使用残差结构”这一观点是不完全正确的。

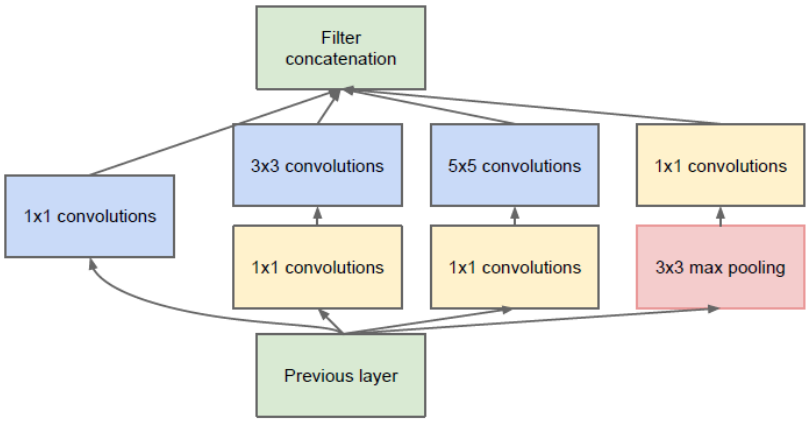


图 2.4 GoogleNet 中的分支结构

ResNet[49]通过将网络深度增加到 152 层，取得了 ILSVRC2015 比赛冠军。

并通过在 cifar 数据集上将 110 层网络与 1202 层网络进行比较，得到虽然训练误差相近，但是后者由于过拟合在测试集上表现略差的实验结果，证明了深度的增加并不总是带来分类准确率的提升，相反在超出一定深度后性能开始下降。ResNet-152 权重数量约为 VGG-19 的一半，基本组成结构如图所示，其中 x 是输入，H(x)是基本组成单元的期望映射输出，F(x)是新增卷积层的期望拟合函数，则当 F(x)=H(x)-x 时，基本组成单元完美拟合了 H(x)；当 F(x)=0 时，网络退化为其较浅版本。通过使用残差跳接的方式，使网络的优化目标由原来的拟合输出 H(x)变成拟合输出和输入的差 H(x)-x，有效缓解了训练非常深的网络时梯度消失的问题，并且相较于无跳接的网络表现出了更快的收敛速度。

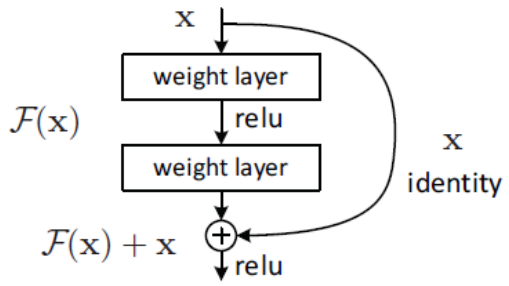


图 2.5 ResNet 结构图

NasNet[50]由 Google 在 2017 年提出，在 ImageNet 图像分类验证集上准确率达 82.7%，是目前为止性能最好的图像分类网络。NasNet 借鉴了 ResNet 和 GoogleNet 重复堆叠思想，通过堆叠基本单元来构建整个网络结构。首先在 cifar-10 使用 RNN 控制器用来预测卷积单元和降维单元，搜索最佳网络架构；然后利用迁移学习将生成的网络迁移到 ImageNet 和 COCO 数据集上。此外，NasNet 还可以设置大小，以非常低的计算成本取得了良好的准确性。

本质上来说，卷积神经网络由许多滤波器组成。前两层的滤波器或对某种颜色敏感，或对某种形状，如直线、圆，可以看做是从图片中提取了某种频率分量的信号。随着网络的层数增加，高层波滤器的输出可以看做是对一系列规则的响应，比如输入特征向量在“该位置出现了六边形”的规则下给出了 1 的取值，那么输出特征图中可能有一组向量编码了“图中有一个蜂窝结构的物体”的信息。

2.3 循环神经网络

循环神经网络是一种通过隐藏层节点周期性的连接，来捕捉序列化数据中动态信息的神经网络，广泛应用于和序列有关的场景，如一帧帧图片组成的视频，一个个片段组成的音频，和一个个单词组成的句子。与其它前向神经网络不同，RNN 将上下文信息保存在网络的内部状态中，能够在任意长的上下文窗口中存储、学习、表达相关信息，而且不再局限于传统神经网络在空间上的边界，可以在时

间序列上有延拓。尽管 RNN 有一些传统的缺点，如难以训练，参数较多，但近些年来关于网络结构、优化手段和并行计算的深入研究使得大规模学习算法成为可能，尤其是 LSTM 与 BRNN 算法的成熟，使得图像标注、手写识别、机器翻译等应用取得了突破性进展。

2.3.1 RNN

RNN[51]在 1982 年由 Hopfiled 提出，其网络结构如图所示。

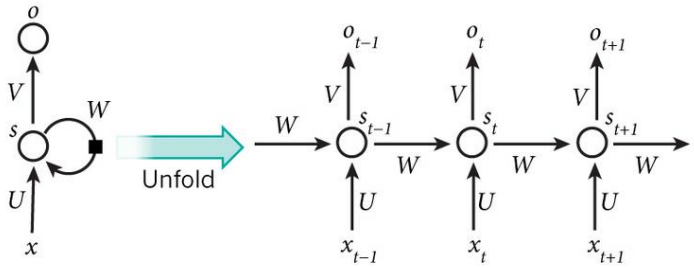


图 2.6 NN 网络结构图

其中 U、V、W 是待优化的权重，x 是输入，s 是隐层状态，o 是输出。t 时刻的输出 o_t 不仅由 t 时刻的输入 x_t 决定，而且通过 W 由 t 时刻的隐层状态 s_t 决定，进而与历史输入相关联。数学表述为

$$\begin{aligned} s_t &= \text{sigmoid}(Ws_{t-1} + Ux_t + b_s) \\ o_t &= \tanh(Vs_t + b_o) \end{aligned} \tag{2.17}$$

2.3.2 BRNN

BRNN(Bi-directional RNN)[52]由 Schuster 在 1997 年提出，其基本思想是 t 时刻的输出不仅依赖于序列之前的元素，也跟 t 时刻之后的元素有关。结构上由两个方向相反的 RNN 构成，这两个 RNN 连接着同一个输出层，这就达到了同时获取输入序列中每一个点的完整的过去和未来的上下文信息的目的。具体结构如图所示。

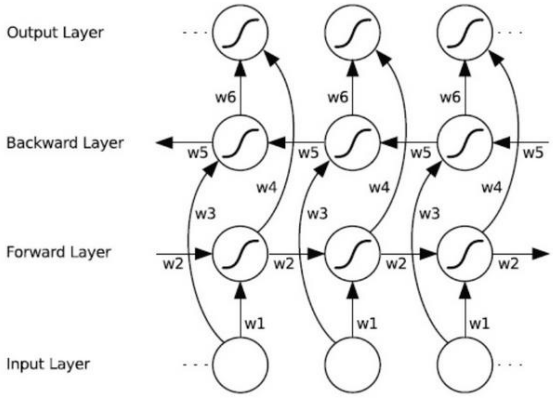


图 2.7 BRNN 结构图

2.3.3 LSTM

传统的 RNN 是从输入的角度上看，非常深的神经网络，即 t 时刻的输出不仅由 t 时刻的输入决定，而且通过状态变量由 t-1, t-2,…,0 时刻的输入共同决定。t 时刻的输出产生的梯度，在通过 BPTT(BackPropagation Through Time)[53]向前传播以更新与 0 时刻输入相关联的 w 时，要经过一个 sigmoid 激活函数（导数取值 0~0.25）和多个 tanh 激活函数（导数取值 0~1），而激活函数的取值很容易饱和，在取值饱和的地方梯度值会非常小，在给定学习率的情况下，梯度可能发生剧烈变化，从而产生梯度爆炸和梯度消失的问题，导致网络难以训练。

LSTM[54]是 RNN 的一个变种，属于反馈神经网络的范畴。理解 LSTM 的关键在于把握c_t(memory cell)的变化。t-1 时刻的记忆c_{t-1}通过遗忘门丢掉一些历史信息，通过输入门添加一些当前信息，得到c_t。然后让c_t通过输出门得到h_t(hidden state)。其结构如图所示

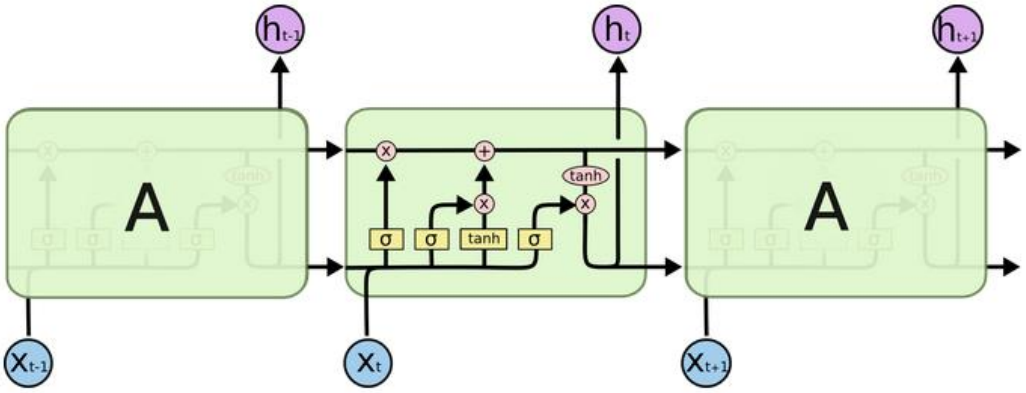


图 2.8 LSTM 结构图

遗忘门、输入门、输出门的输出由前一时刻的输出和当前时刻的输入决定，定义为

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t] + b_f) \tag{2.18}$$

$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t] + b_i) \tag{2.19}$$

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_t] + b_o) \tag{2.20}$$

c_t和h_t的更新公式为

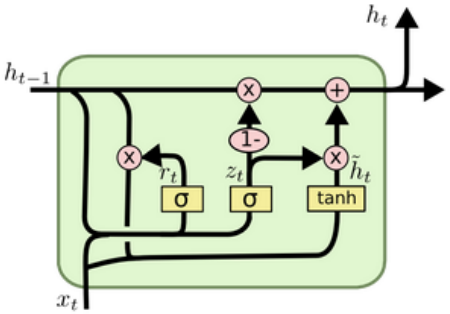
$$c_t = f_t * c_{t-1} + i_t * \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{2.21}$$

$$h_t = o_t * \tanh(c_t) \tag{2.22}$$

通过加入这些门控单元，LSTM 的梯度得以以常数级别向后传播，不会因非线性激活函数而导致梯度以指数级衰减，解决了传统的 RNN 由于梯度爆炸和梯度消失导致训练效率低下甚至无法训练的问题，相较于标准 RNN 具有较长期的短期记忆。

2.3.4 GRU

GRU(Gated Recurrent Unit)[55]是 LSTM 最流行的一个变体，它将 LSTM 中的遗忘门和输入门结合起来作为更新门，并将 memory cell 和 hidden state 进行了合并，最终获得了 LSTM 的简化版本，但在大多数任务中其表现与 LSTM 不相伯仲，因此也成为了常用的 RNN 算法之一。其结构如图所示。



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

图 2.9 GRU 结构图

2.3 注意力机制

人类的感知有一个重要的特性是不会立即处理外界的全部输入，而会将注意力集中于所选择的某部分来获取所需要的信息，然后结合不同时间段的局部信息来建立一个内部的场景表示，从而引导眼球的移动及做出决策[56]。注意力机制的基本思想是在人类的视觉感知中，虽然看到了整幅画面，但在特定时刻，意识只集中在画面中的某一个部分上，其它部分虽然也被眼球捕捉到，但是由于分配给它们的注意力资源很少而被忽略。自 2014 年由 google mind[57]在 RNN 模型上使用了注意力机制来进行图像分类后，该机制广泛应用于机器翻译[58,59]、自动摘要生成[60,61]、问答系统[62,63]、文本分类[64-66]、图片生成[67]、图像分类[68]、图像语义分析[17,69-71]等任务中。

[17]首次使用注意力机制来学习生成描述过程中应该注意图片中的哪一区域，并取得了明显的模型性能提升。因此不再使用 CNN 经过了全连接层的特征，而是使用前面卷积层的特征。维度为 w*h*channel=14*14*512 的特征图，构成了特征集合 $a=\{a_1,a_2,...,a_L\}, a_i \in R^D$，L=w*h，D=channel，$a_i$ 代表了从原图的某个子区域抽取出的具有语义信息的特征。通过使用注意力机制，每次只注意特征集合 a 中的某几个 a_i。具体实现中，使 LSTM 中的三个门控信号不仅由前一时刻的输出和当前时刻的输入决定，而且由当前的上下文信号，即 a_i 的加权平均决定。即

$$f_t = \sigma(W_f \bullet [h_{t-1}, x_t, z_t] + b_f) \tag{2.23}$$
$$i_t = \sigma(W_i \bullet [h_{t-1}, x_t, z_t] + b_i) \tag{2.24}$$

$$o_t = \sigma(W_o \bullet [h_{t-1}, x_t, z_t] + b_o) \tag{2.25}$$

而 memory cell 和 hidden state 分别通过下式更新

$$\begin{aligned} c_t &= f_t * c_{t-1} + i_t * \tanh(W_c[h_{t-1}, x_t, z_t] + b_c) \\ h_t &= o_t * \tanh(c_t) \end{aligned} \tag{2.26}$$

其中 x_t 是当前时刻的输入，即上一时刻的输出字符对应的词向量编码。z_t 作为上下文向量，保存了输入图像中某一特定位置的视觉信息，定义为

$$z_t = \sum_{i=1}^{L} w_{it} a_i \tag{2.27}$$

其中 a_i 是子图的特征向量，w_{it} 是 a_i 在 t 时刻对应权重，代表根据已经生成了哪些单词来决定当前时刻要看哪些图像特征。初始记忆状态与单元状态分别通过各子图特征向量的平均计算，即

$$c_0 = f_{init,c}(\frac{1}{L}\sum_{i=1}^{L} a_i) \tag{2.28}$$

$$h_0 = f_{init,h}(\frac{1}{L}\sum_{i=1}^{L} a_i) \tag{2.29}$$

$f_{init,c}$、$f_{init,h}$ 采用多层感知机模型，即一个或多个全连接层实现。

硬注意力模型通过使用最大似然估计和蒙特卡洛方法来进行优化训练。w_{it} 是一个 onehot 向量，只有被选中的区域对应的权重为 1，其它为 0。假设 w_{it} 符合参数为 α_{ti} 的多元贝努利分布，即

$$p(s_{t,i} = 1 | s_{j<t}, a) = \alpha_{ti} \tag{2.30}$$

$$\alpha_{ti} = soft\max(e_{it}) \tag{2.31}$$

$$e_{it} = f_{att}(a_i, h_{t-1}) \tag{2.32}$$

软注意力模型下，$\sum_i w_{it} = 1$，w_{it} 定义为

$$w_{it} = soft\max(e_{it}) \tag{2.33}$$

$$e_{it} = f_{att}(a_i, h_{t-1}) \tag{2.34}$$

$$f_{att}(a_i, h_{t-1}) = \tanh(a_i + h_{t-1} w_h) \mathrm{u}_{att} \tag{2.35}$$

其中 w_h 将隐层状态 h_{t-1} 投影到图像特征编码空间，u_{att} 与 a_i 为长度相同的一维向量，最终得到常量 e_{it}。为了让图像的每个区域在整个解码过程中的权重之和都相等，从而图像的各个区域都对生成描述起到贡献，损失函数定义为

$$L_d = -\log(P(y|x)) + \lambda \sum_{i=1}^{L} (1 - \sum_{t=1}^{T} \alpha_{ti})^2 \tag{2.36}$$

由于注意力向量是通过 softmax 激活函数生成的，即每一时刻注意力权重之和是 1，因此为软注意力模型添加了门控信号 $\beta_t = \sigma(f_\beta(h_{t-1}))$，并使得 t 时刻生成的图形上下文信号 z_t 通过该门控信号后，再和当前输入单词的词向量编码一起输入 LSTM，即 $\hat{z}_t = \beta_t z_t$。添加该机制后，模型的注意力更多地集中在了图中的物体上。

[70]在使用卷积神经网络提取图像特征环节中加入了注意力机制。每一张特征图本质上是对一系列滤波器产生的响应，那么根据当前上下文，选取相关图像

特征将有助于改善模型性能。比如当需要预测 cake 这个单词时，为含有蛋糕、火焰、灯光和蜡烛形状的特征图分配更大的权重有助于特征抽提。由于每张特征图依赖于前一层的特征图，因此对多层特征图使用注意力机制便于获取多层语义抽象概念。

[71]设计了基于视觉哨兵自适应注意力模型，在生成每个单词时，模型决定是将注意力放到图片中的某个区域还是注意视觉哨兵，当语言模型不能提供有效信息的时候，模型才再次注意到图片。在[17]将图像上下文作为 LSTM 的输入的基础上，将图像上下文与 LSTM 的 hidden state 一同送入语句生成 MLP 模型。基于注意力机制的 LSTM 模型结构如图所示。

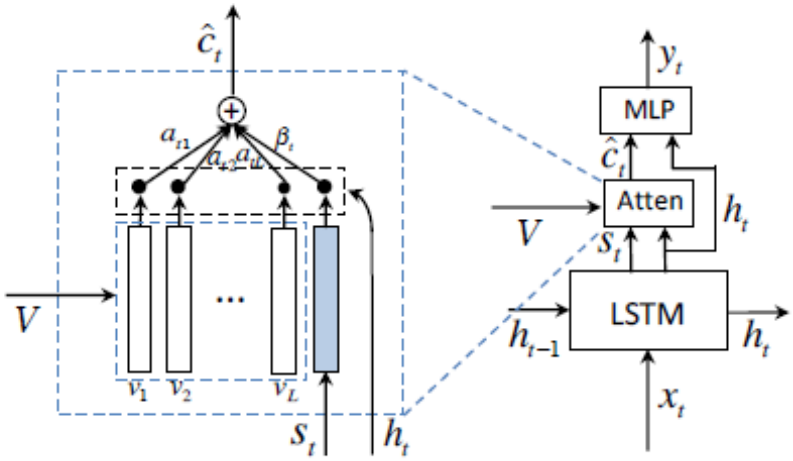


图 2.10 基于视觉哨兵自适应注意力模型

图像上下文的生成不再根据 h_{t-1} 确定，而是由当前时刻生成的 h_t 决定。与 ResNet 类似，生成的图像上下文信号 c_t 可以看做是 MLP 模型所需要的输入与 h_t 之间的残差。实验表明该方法与[17]相比，在 BLEU 等评价指标上具有更好的表现。除此之外，设计了视觉哨兵信号 s_t 及门信号 β_t，最终作为残差补充输入 MLP 中的上下文信号 $\hat{c}_t$ 被定义为

$$\hat{c}_t = \beta_t s_t + (1-\beta_t)c_t \tag{2.37}$$

$$s_t = \sigma(W_x x_t + W_h h_{t-1}) \odot \tanh(c_t) \tag{2.38}$$

其中 x_t 是 t 时刻 LSTM 的输入，c_t 为 LSTM 的 memory cell。而 β_t 通过在图像特征集合 $\{v_1, v_2, ..., v_L\}$ 中添加 s_t，得到

$$\hat{\alpha}_t = soft\max(w_h^T \tanh(W_{v,s_t}[V_t; s_t] + W_g h_t)) \tag{2.39}$$

然后取向量 $\hat{\alpha}_t \in R^{L+1}$ 中最后一个值，即 $\beta_t = \hat{\alpha}_t[L+1]$。最后在给定输入单词的条件下，单词表上的输出概率分布通过式(2.40)计算得出，即

$$p_t = soft\max(W_p(\hat{c}_t + h_t)) \tag{2.40}$$

通过上述机制，成功训练出了一个在生成名词，如“dishes”，“people”，“cat”,或形容词，如“giant”，“metal”，“yellow”时关注图片中的相应

区域，而在生成“the”，“of”，“to”等词时不参考图片，即知道什么时候去关注图像特征的模型。

[72]设计了基于区域的注意力模型，模型结构如图所示。t 时刻视觉上下文信号的生成不仅由 t 时刻的输入、LSTM 在 t-1 时刻更新的隐层状态和基于区域的图像特征向量决定，而且由前一时刻生成的视觉上下文信号决定。通过使用单隐层的神经网络，为每个区域特征生成权重，通过加权平均得到 t 时刻的图像上下文信号。

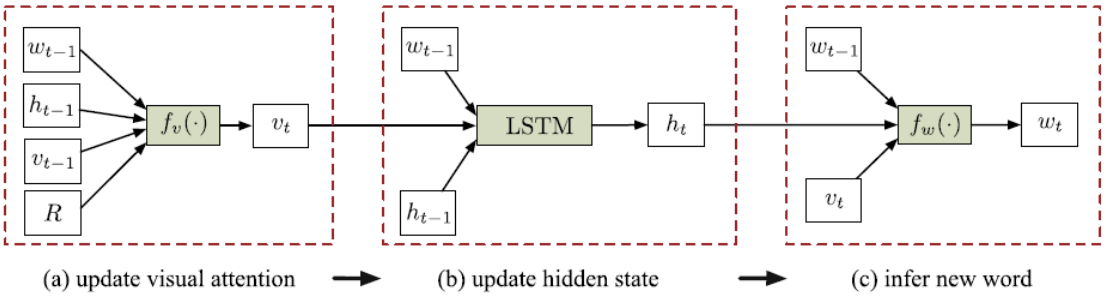


图 2.11 基于区域的注意力模型

[73]设计了基于图像全局特征的注意力模型。首先使用 VGG-16 倒数第二个全卷积层的特征对图像进行编码，然后使用单隐层前馈神经网络生成当前时刻的注意力向量，如式 2.41 所示。

$$h_t^a = \sigma(W_g h_{t-1} + W_y s_{t-1} + a) \tag{2.41}$$

$$a_t = \sigma(W_m h_t^a) \odot a \tag{2.42}$$

其中 h_{t-1} 是 LSTM 上一时刻更新后的隐层状态，s_{t-1} 是上一时刻的输出单词，W_y 是词向量编码矩阵，a 是 VGG-16 倒数第二个全卷积层生成的特征。之后将 a_t 通过线性模型映射到语义特征空间后，和当前时刻的输入单词 x_t 一起提供给 LSTM 进行下一单词的预测，即

$$\begin{aligned} v_t &= W_a a_t + b_a \\ h_t &= f(v_t + x_t + W_h h_{t-1}) \end{aligned} \tag{2.43}$$

3. 图像描述系统设计

3.1 总体规划

系统由基于深度学习的图像描述算法设计部分与 web 应用程序部分构成。其中基于深度学习的图像描述算法是系统的核心部分，该模块需要完成图像特征提取与自然语言描述生成。web 应用程序主要负责与用户交互，其中后台程序接收浏览器用户请求，并对请求做出响应，根据用户上传的图片调用图像描述算法模块，将描述语句返回给用户。

3.2 图像区域特征提取

不同于已有的一些模型，将整张图片编码成一个特征向量，本文使用特征向量集来对图片进行编码。[17]选择VGG-16最后一个卷积层输出的14*14*512的特征图作为图像特征，将特征图均匀分成14*14的区域，在硬注意力机制下每生成一个单词注意一个区域，软注意力机制下每生成一个单词注意各区域的加权平均，然而每个区域对应的语义是模糊的，也无法保证一个区域恰好完整编码了图中的一个物体。更加接近人类视觉感知的方式，首先定位图中的物体，然后对其进行编码，再通过注意力机制使用RNN进行解码，无疑有助于我们进一步探索注意力机制。[72]使用selective search来对图像进行多尺度分割，选最可能出现物体的29个区域连同原图共30个区域，依次送入CNN中提取图像特征向量，然而selective search本身就是耗时的算法，在CPU上处理一张图片需要2s；30个区域依次送入CNN中提取图像特征更是存在运算资源的浪费。而且由于CNN的输入图片大小固定，需要将候选区域强行缩放到要求大小，引入了几何畸变。近年来随着目标检测技术的发展，Faster R-CNN模型被提出，通过使候选区域生成与区域特征提取在同一神经网络中进行，提高了参数共享程度，将候选区域生成的时间缩短到10ms。在使用VGG-16完成目标检测的全部流程时，GPU上达到了5帧每秒。模型示意图如图3.1所示，主要在Fast R-CNN模型中加入了RPN(Region Proposal Network)来生成候选区域。之后根据生成的候选区域在特征图上做RoI(Region of Interest) pooling，生成固定长度的特征向量，最后送入分类器获取物体类别。

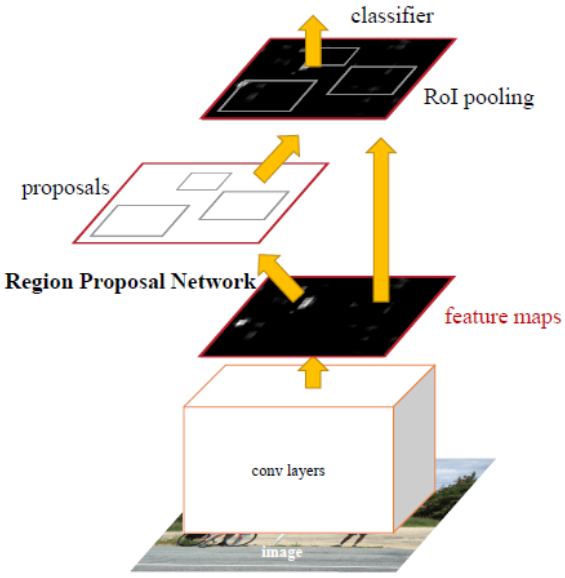


图 3.1 Faster RCNN 模型示意图

RPN 通过在原本的特征图上添加额外的卷积层实现，可以高效地发现不同大

小和长宽比的物体。模型采用了全卷积网络，因而输入可以是任意大小的特征图。选择滑动窗口大小为 3*3，通过在共享卷积层的最后一层移动 RPN，在对应位置上输出为一系列标有置信度的矩形框。网络结构如图所示。首先在滑动窗口的中央预设 k 个具有不同尺度和长宽比的锚点(anchor box)以检测不同尺度和长宽比的物体。通过取 $\text{scale}=\{128,\ 256,\ 512\}$， $\text{ratio}=\{0.5,1,\ 2\}$，对于 14*14 的输入特征图，一共生成 1764 个锚点。然后回归层(reg layer)通过线性回归，在每个锚点上预测一个矩形框。由于矩形框具有(x, y, w, h)四个参数，因而回归层的输出结果为 4k 大小。分类层则预测在该锚点上生成的矩形框的置信度。由于使用了两类的 softmax 层来做分类，因此同时给出有物体的概率与无物体的概率，即输出 2k 个结果。网络的损失函数的定义为

$$L(\{p_i\},\{t_i\})=\frac{1}{N_{cls}}\sum_i L_{cls}(p_i,p_i^*)+\lambda\frac{1}{N_{reg}}\sum_i p_i^* L_{reg}(t_i,t_i^*) \tag{3.1}$$

其中 p_i 和 p_i^* 分别代表第 i 个锚点上预测的置信度与真实值。当锚点与图中物体的外接矩形框重合度达 0.7 以上，或与某个物体的外接矩形框的重合度最高时，$p_i^*=1$；当锚点与图中所有物体的重合度均小于 0.3 时，$p_i^*=0$；上述两种情况都不满足的锚点不计入损失函数。由于一张图片上 $p_i^*=0$ 的锚点远远多于 $p_i^*=1$ 的锚点，因此在每个批次的训练中随机采样 256 个锚点，并且保证正负锚点之比为 1:1。t_i 和 t_i^* 分别代表预测矩形框与真实矩形框的位置。$p_i^* L_{reg}(t_i,t_i^*)$ 表明只对 $p_i^*=1$ 的锚点计算回归损失。$L_{reg}(t_i,t_i^*)$ 选择平滑 L1 损失，定义为

$$L_{reg}(t_i,t_i^*)=\sum_{i\in\{x,y,w,h\}} smooth_{L_1}(t_i-t_i^*) \tag{3.2}$$

而

$$smooth_{L_1}(x)=\begin{cases}0.5x^2 & if\ |x|<1\\ |x|-0.5 & otherwist\end{cases} \tag{3.3}$$

这样当二者相差太多，梯度为±1，可以避免发生梯度爆炸，使训练过程更加稳定。λ 用于平衡 cls 与 reg 项之间的权重，当 λ=1时，二者重要性相当。N_{cls} 与 N_{reg} 是两个归一化参数，分别取批次大小和锚点数目。

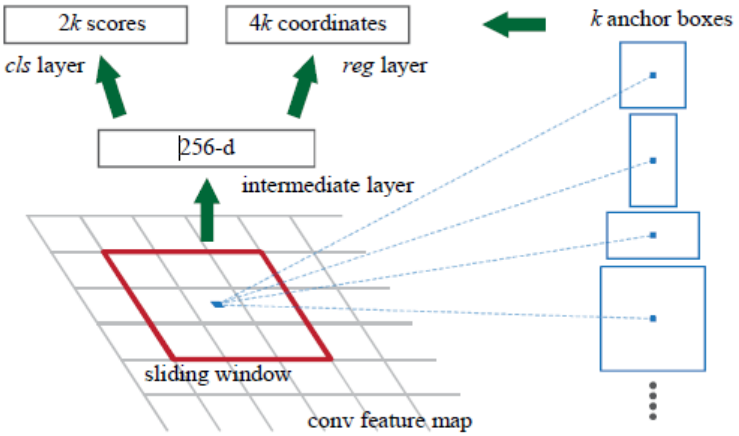


图 3.2 Faster R-CNN 锚点

为了使损失函数不受图片大小的影响，保证训练过程能够正常收敛，在进行外接矩形框回归时，对绝对坐标进行变换

$$
\begin{aligned}
t_x &= \frac{x - x_a}{w_a} \quad t_y = \frac{y - y_a}{h_a} \\
t_w &= \log\frac{w}{w_a} \quad t_h = \log\frac{h}{h_a} \\
t_x^* &= \frac{x^* - x_a}{w_a} \quad t_y^* = \frac{y^* - y_a}{h_a} \\
t_w^* &= \log\frac{w^*}{w_a} \quad t_h^* = \log\frac{h^*}{h_a}
\end{aligned}
\tag{3.4}
$$

其中 x， y， w 和 h 分别代表矩形框的中心点和长宽。 x， x_a 和 x^* 分别代表预测矩形框，锚点，真实矩形框。

通过 RPN 在原图上初步获得候选区域的位置后，将边框的(x,y,w,h)坐标相对于图片大小，缩放到(0,1)之间，根据该相对坐标，在特征图上裁取对应位置。并将裁切下来的特征图缩放到固定大小，最后进行 max_pooling,即得到 RPN 给出的候选区域的特征向量。值得一提的是尽管到了卷积网络的高层，如 VGG-16 的 conv5-3 层，特征图的感受野已经到了 196*196,但是特征图与原图在空间上依旧存在对应关系。如图所示，第 55 个滤波器的箭头所指位置在图中出现圆形时的响应最强，并且车轮所处位置恰好就是对圆形敏感的神经元在特征图上的位置。即便在网络中，一开始图片和特征图的空间对应关系并不明显，也可以在后续的训练中，通过不断调整权重，使得图中相邻的区域的特征，在特征图上也相邻。

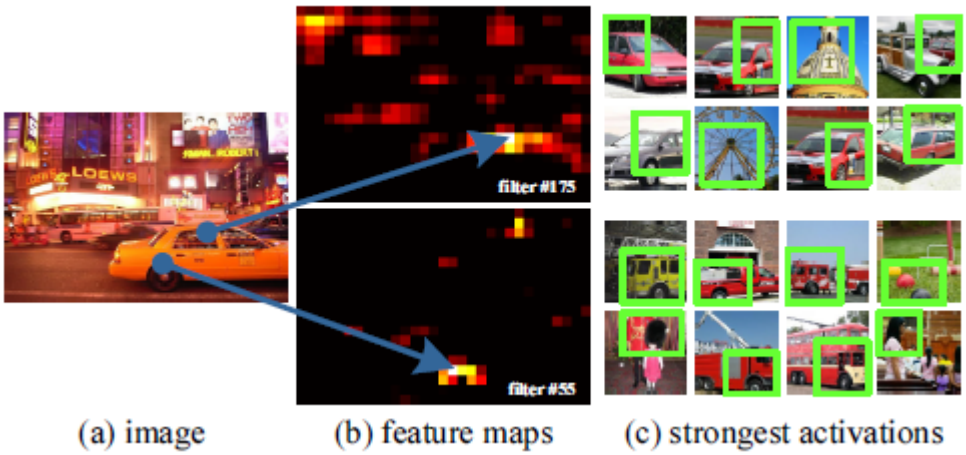


图 3.3 Faster R-CNN 特征图映射

Faster R-CNN 采用交替训练 RPN 和 Fast R-CNN 的方式来获取网络权重。首先采用在 Image Net 上预训练的权重进行网络初始化后，单独对 RPN 进行调优。然后依旧采用在 Image Net 上预训练的权重进行网络初始化，并使用 RPN 产生的候选区域单独训练 Fast R-CNN。此时 RPN 和 Fast R-CNN 具有各自的卷积特征提

取层。再把 Fast R-CNN 的卷积特征提取层权重赋给 RPN，并在之后的训练中不再更新，只对 RPN 的区域建议层进行调优。最后保持 RPN 的区域建议层权重固定，单独优化 Fast R-CNN 的权重。

本文使用 Faster R-CNN 模型提取图片特征。RPN 生成的候选区域很多，通过使用非最大抑制，即如果两个区域之间的 IoU 大于某一阈值，舍弃得分低的那个，最终保留得分最高的 100 个区域，通过 roi pooling 共生成 100*2048 维图像特征。相较于当前直接使用 VGG 或 ResNet 等用于分类卷积神经网络来进行全局图像特征提取，Faster R-CNN 设计了锚点来检测不同尺度的物体。由于候选区域生成模型是通过 k 个锚点对应的 k 个回归器实现的，每个回归器负责检测不同大小和长宽比的物体，因此尽管输入特征的大小或尺度是固定的，不同大小或长宽比的物体依旧可以被有效检测出来。进而保证了提取的图像区域特征中，保留了图中较小物体，如“sink”，“surfboard”，“clock”和“frisbee”等的信息，有效改善了生成其对应描述时模型表现得不尽如人意的问题。除此之外，VGG 等网络因为具有全连接层，需要固定大小的图片输入，例如 224*224，而实际中需要处理的图片往往具有不同大小，因此需要对其进行裁切或拉伸，进而带来了图片信息丢失或引入不希望的几何畸变的问题。Faster R-CNN 或 Fast R-CNN，在卷积层和全连接层之间插入了 RoI pooling 层，而 RoI pooling 层会将输入特征图通过最大池化映射到固定大小的输出，使得网络可以接受任意大小的图片作为输入，提取的图片特征更为可靠。

3.3 LSTM 生成描述语句

基于注意力机制的 LSTM 模型实现流程如图所示。LSTM 的输入是当前单词的词向量编码及图像上下文，hidden state 和 memory cell 大小相同，更新公式为

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ g_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} T \begin{pmatrix} P_w w_{t-1} \\ h_{t-1} \\ v_t \end{pmatrix} \tag{3.5}$$

其中 i_t， f_t 和 c_t 分别代表输入、遗忘和输出门。P_w 是词向量编码矩阵，v_t 是当前图像上下文信号，T 代表仿射变换。LSTM 的 memory cell 和 hidden state 通过式(4.12)给出

$$\begin{aligned} c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\ h_t &= o_t \odot \tanh(c_t) \end{aligned} \tag{3.6}$$

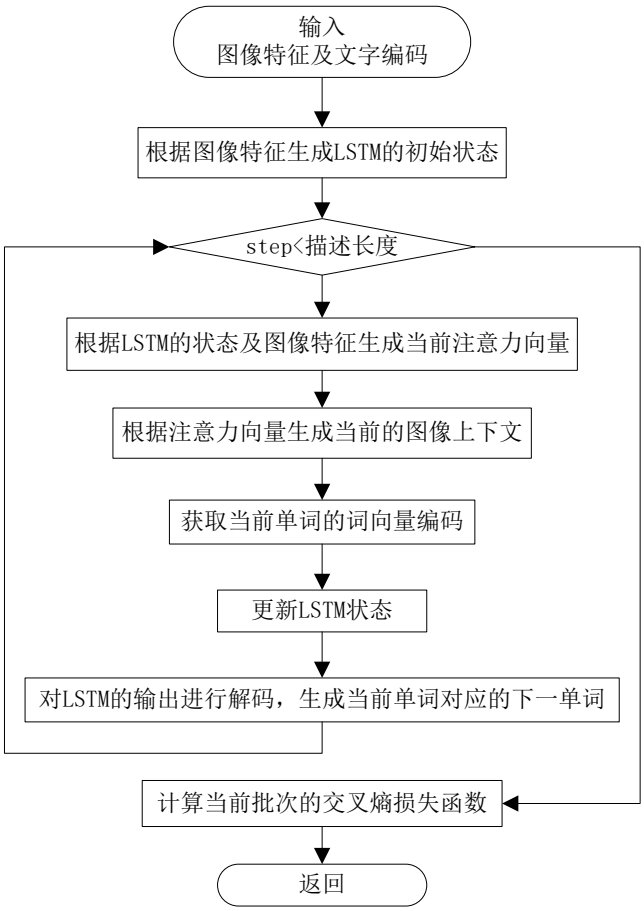


图 3.4 描述语句生成流程

LSTM 初始状态的生成采用了多层感知机模型，其中隐层神经元的大小控制着模型复杂度和参数数量。而隐层神经元的数目不是越多越好。隐层神经元数量增加会使得神经网络的训练难度增大，消耗更多计算时间，并可能导致模型过拟合。初始状态的计算公式如下所示。

$$\begin{aligned} a_{mean} &= \frac{1}{L}\sum_{i=1}^{L} a_i \\ c_0 &= W_{o-c_0}\tanh(W_{h-c_0}a_{mean} + b_{h-c_0}) + b_{o-c_0} \\ h_0 &= W_{o-h_0}\tanh(W_{h-h_0}a_{mean} + b_{h-h_0}) + b_{o-h_0} \end{aligned} \tag{3.7}$$

其中$\{a_1, a_2, ..., a_L\}$是卷积神经网络提取的特征，当使用 VGG-19 的 5-3 卷积层特征时，L=14*14，$a_i \in R^{512}$。当多层感知机模型中，隐层神经元数目为 512，LSTM 的隐层神经元数目取 1024 时，参数的维度分别为$W_{h-c_0}, W_{h-h_0} \in R^{512\times512}$，$b_{h-c_0}, b_{h-h_0} \in R^{512}$，$W_{o-c_0}, W_{o-h_0} \in R^{1024*512}$，$b_{o-c_0}, b_{o-h_0} \in R^{512}$。

对 LSTM 的输出进行解码同样采用多层感知机模型，计算公式如下所示。

$$\begin{aligned} o_t &= [h_t, a_t, y_t] \\ p_t &= W_p\tanh(W_o o_t + b_o) + b_p \end{aligned} \tag{3.8}$$

其中h_t是 LSTM 的输出，a_t是当前图像上下文，y_t是当前输入单词的词向量编码。当多层感知机模型中，隐层神经元数目为 1024，字典大小为 5000，词向量编码空间大小为 512 时，$h_t \in R^{1024}$，$a_t \in R^{512}$，$y_t \in R^{512}$，参数的维度分别为

$W_o \in R^{1024\times2048}$，$b_o \in R^{1024}$，$W_p \in R^{5000*1024}$，$b_p \in R^{5000}$。

3.4 注意力机制

传统的注意力机制要求在生成每个词的时候，都为每个区域特征生成不同权重，将加权平均后的图像特征输入 LSTM 中进行解码。例如图片的描述为"A red bus driving down a street"，在生成" A red bus"时，注意力均集中在图中巴士所在区域。但是在生成"A"时，区域特征已经提供给了解码 LSTM，在生成"red"和"bus"时，如果依旧将巴士所在的区域特征提供给 LSTM，一方面造成特征的重复输入，给 LSTM 的解码过程造成干扰，另一方面为注意力模型根据 LSTM 的隐层状态确定当下注意力所在位置带来困难，注意力可能离开巴士所在区域。事实上，NIC 模型的成功表明图像特征对应着一组描述词，而不是一个。[71]设计了基于视觉哨兵自适应注意力模型，在生成每个单词时，模型决定是依据图片特征向量还是视觉哨兵，当语言模型需要进一步提供视觉信息时，注意力机制才为相应图片区域分配权重。但是在生成每个单词时，模型要么参考图片，要么参考上一时刻的隐层状态，可能导致特征的冗余输入，并使得模型复杂度增加。

本文对注意力机制进行改进，使得 LSTM 在生成每个单词时不再总是需要参考图片特征，注意力向量生成公式如下所示。

$$e_i = w_{e_i}(\tanh(w_{a_i}a_i + w_c c_t + \mathrm{w}_{e_i}e_i + b_c))$$
$$\alpha_i = \frac{\exp(e_i)}{\sum_L \exp(e_i) + b_e} \tag{3.9}$$

其中$\{a_1, a_2, ..., a_L\}$是卷积神经网络提取的特征，当使用 VGG-19 的 5-3 卷积层特征时，L=14*14，$a_i \in R^{512}$。α=$[\alpha_1, \alpha_2, ..., \alpha_L]$是对应的注意力向量。注意力向量的生成采用了多层感知机模型，当隐层神经元数目为 512，LSTM 的隐层神经元数目取 1024 时，参数的维度分别为$W_e \in R^{512}$，$W_a \in R^{512\times512}$，$W_h \in R^{512\times1024}$，$b_a, b_h \in R^{512}$。$b_e$ 的存在使得 LSTM 在生成下一单词时不再总是需要参考图像特征，可作为一个偏差量，通过反向传播确定；也可与注意力向量的生成类似，每次根据提取的图片特征和 LSTM 的隐层状态计算得到，即$b_{e_i} = w_{b_{e_i}}\tanh(a_i + w_c c_t + \mathrm{b}_{b_{e_i}})$。本文将对上述两种方式进行实验并进行比较。

3.5 beam search

在预测阶段，生成描述语句时，模型接受描述起始符，输出字典中的每个词是第一个词的概率，如果采用贪婪的思想，选取概率最大的那个词作为当前输出，然后把这个词作为预测第二个词的输入再送入网络，如此循环，直到模型输出结束符，那么这句话就输出完毕。Beam Search 是这种方式的扩展，不再选择概率

最大的词作为当前输出，而是选择当前概率最大的前 k 个词作为候选输出。然后将每个候选词依次输入模型，选择概率最大的前 k 组词作为截至此时的最终答案。最终选择得分最高的句子作为输出。例如对于大小为 3，内容为{a，b，c }的字典，选择 beam size 为 2。生成第 1 个词的时候，选择概率最大的 2 个词，假设为{a,c}；生成第 2 个词的时候，我们将当前序列 a 和 c，分别与词表中的所有词进行组合，得到新的 6 个序列{aa，ab，ac，ca，cb，cc},然后从其中选择 2 个得分最高的作为当前序列，假设为{aa，cb}；之后不断重复这一过程，直到生成结束符为止。最终输出得分最高的序列。这种搜索机制能让模型生成更合适的描述语句。其中 beam size 控制着搜索的复杂度，当 beam size=1 时，即为通常的贪婪算法，而 beam size 越大标志着搜索空间越大，所需时间越长。

4. 实验与分析

4.1 实验数据

到目前为止，深度神经网络模型因其参数数量庞大，是一种需要大量样本来进行训练的模型。样本数量不足会导致过拟合现象，甚至导向错误的结论，因此数据对于基于深度学习的算法至关重要。本文采用 COCO[76]、Flickr8K 和 30K 图像标注数据集对算法进行验证。COCO Caption 数据集由微软推出，包括 82,783 张图片作为训练集，40,504 张图片作为验证集和 40,775 张图片作为测试集。包含了 c5 和 c40 两个数据集：MS COCO c5 的训练集、验证集和测试集图像和原始的 MS COCO 数据库是一致的，每张图片配有 5 个人工标注语句；MS COCO c40 包含 5000 张从 MS COCO 数据集的测试集中随机选出的图片，每张图片对应 40 个人工标注语句。Flickr8K 和 Flickr30K 数据集的图片来源是雅虎的相册网站 Flickr，图片数量分别是 8,000 张和 31,783 张，每张图像对应 5 句人工标注，大多展示的是人们正在参与某项活动的情景。

4.2 评价指标

评价指标对于图像语义理解算法的研究和发展具有重要意义，通过评测可以得知算法存在的问题而不断改进。

BLEU[77](bilingual evaluation understudy)是机器翻译中衡量机器译文与其对应的几个参考译文之间相似度的指标，得分越高说明机器翻译得越好。它方便、快速、结果与人类评价的相关性高，但是因为未考虑语法上的准确性，测评精度会受常用词的干扰，短译句的测评精度有时会较高，没有考虑同义词或相似表达的情况，可能会导致合理翻译被否定，因此 BLUE 指标的得分增长不一定

代表翻译水平的提高。BLUE 采用了 N-gram 的匹配规则，1-gram 描述译文的忠实度，代表机器译文中有多少个词出现在参考译文中；2-gram、3-gram、4-gram 描述译文的流畅度，代表机器译文中有多少连续的 n 个词出现在参考译文中。由于 N-gram 存在只翻译出了部分句子时的匹配度依然会很高的问题，BLEU 在最后的评分结果中引入了长度惩罚因子 BP(Brevity Penalty)。当机器译文长度大于参考译文时，惩罚系数为 1，意味着不惩罚；而当机器译文长度小于参考译文会计算惩罚因子。由于各 N-gram 统计量的精度得分随着阶数的升高而呈指数下降，因此对其采用几何加权平均，权重服从均匀分布，再乘以长度惩罚因子，得到最后的评价公式：

$$BLEU = BP \times \exp(\sum_{n=1}^{N} \frac{1}{N} \log p_n) \tag{4.1}$$

N 取 1、2、3、4，对应 BLEU-1、BLEU-2、BLEU-3、BLEU-4。多元精度得分（n-gram precision scoring）定义为

$$p_n = \frac{\sum_{C \in \{Candidates\}} \sum_{n-gram \in C} Count_{clip}(n-gram)}{\sum_{C' \in \{Candidates\}} \sum_{n-gram' \in C'} Count(n-gram')} \tag{4.2}$$

Candidates 代表机器译文中句子。$Count(n-gram)$ 指在该句中该 $n-gram$ 的出现次数，$Count_{clip}(n-gram)$ 指将 $Count(n-gram)$ 截断到在参考译文中最大出现次数。惩罚因子定义为

$$BP = \begin{cases} 1 & c > r \\ e^{(1-r/c)} & c <= r \end{cases} \tag{4.3}$$

c 代表机器译文的长度，r 表示参考答案的有效长度，当存在多个参考译文时，选取和翻译译文最接近的长度。例如对于机器译文中的一句“The cat isin the mat”,参考译文 1“The cat is on the rubber mat”,参考译文 2“There is a cat on the rubber mat”。首先计算 p_1。

词（1-gram）	机器译文	参考译文 1	参考译文 2	$Count_{clip}(n-gram)$
The	2	2	1	2
cat	1	1	1	1
is	1	1	1	1
in	1	0	0	0
mat	1	1	1	1

即 $\sum_{1-gram' \in C'} Count(1-gram') = 6$，$\sum_{1-gram \in C} Count_{clip}(1-gram) = 5$，$p_1 = \frac{5}{6}$。对于 p_2

词（2-gram）	机器译文	参考译文 1	参考译文 2	$Count_{clip}(n-gram)$
Thecat	1	1	0	1
cat is	1	1	0	1
is in	1	0	0	0
in the	1	0	0	0

the mat	1	0	0	0

即 $\sum_{2-gram' \in C'} Count(2-gram') = 5$，$\sum_{2-gram \in C} Count_{clip}(2-gram) = 2$，$p_2 = \frac{2}{5}$。对于 p_3

词（3-gram）	机器译文	参考译文 1	参考译文 2	$Count_{clip}(n-gram)$
Thecatis	1	1	0	1
cat is in	1	0	0	0
is in the	1	0	0	0
in the mat	1	0	0	0

即 $\sum_{3-gram' \in C'} Count(3-gram') = 4$，$\sum_{3-gram \in C} Count_{clip}(3-gram) = 1$，$p_3 = \frac{1}{4}$。对于 p_4

词（4-gram）	机器译文	参考译文 1	参考译文 2	$Count_{clip}(n-gram)$
Thecatis in	1	0	0	0
cat is in the	1	0	0	0
is in the mat	1	0	0	0

即 $\sum_{4-gram' \in C'} Count(4-gram') = 3$，$\sum_{4-gram \in C} Count_{clip}(4-gram) = 0$，$p_3 = 0$。长度惩罚因子 $BP = e^{(1-7/6)} = 0.8465$。最终可得 BLEU-1=0.7054，BLEU-2=0.4887，BLEU-3=0.3697，BLEU-4=0

METEOR[78](Metric for Evaluation of Translation with Explicit Ordering)同样是机器翻译中基于句子的评价指标，于 2004 年由 Lavir 发现在评价指标中召回率的意义后提出。他们的研究表明，召回率基础上的标准相比于那些单纯基于精度的标准(如 BLEU)，其结果和人工判断的结果更具相关性。它不仅在整个集合，而且在句子和分段级别，也能和人类判断的相关性高。在全集级别，它的相关性是0.964，BLEU 是0.817。在句子级别，它的相关性最高到了0.403。其定义为

$$METEOR = F_{mean}(1-p) \tag{4.4}$$

其中 F_{mean} 是精度和召回率的加权调和平均，定义为

$$\frac{1}{F_{mean}} = \frac{1}{10}\frac{1}{P} + \frac{9}{10}\frac{1}{R} \tag{4.5}$$

$$P = \frac{m}{w_t} \tag{4.6}$$

$$R = \frac{m}{w_r} \tag{4.7}$$

m 指的是机器译文中的单词出现在参考译文中的个数，w_t 是机器译文中单词的个数，w_r 是参考译文中单词的个数。在将机器译文中的单词与参考译文的单词进行匹配的时候，使用了 WordNet 的同义词库，因而 METEOR 具有其它指标没有的同义词匹配的功能。p 是惩罚项，机器译文与参考译文的语序越不一致取值越高，如果机器译文与参考译文没有连续两个词以上的匹配，惩罚项将会把 METEOR 的得分降低一半。定义为

$$p = 0.5(\frac{c}{u_m})^3 \tag{4.8}$$

c是连续有序的块(chunks)的数目，u_m是单词匹配数。当存在匹配单词数相等的两组不同匹配时，采用绝对距离最小的匹配。例如对于机器译文“The cat sat on the mat”，参考译文“on the mat sat the cat”,第1个the与第1个the匹配，因此c=6，u_m=6，p=0.5，m=6，m_t=6，m_r=6，P=R=1，$METEOR$=0.5。而对于机器译文“thecatsatonthemat”与参考译文“thecat was satonthemat”,METEOR则高达0.9654。

ROUGE[79](Recall-Oriented Understudy for Gisting Evaluation)是2004年由ISI的Chin-Yew Lin提出的一系列基于n-gram的评价方法，包括ROUGE-N(N=1、2、3、4，代表1-gram到4-gram)，ROUGE-L，ROUGE-S, ROUGE-W，ROUGE-SU等。现被广泛应用于自动摘要评测任务中。机器翻译中常用的ROUGE-L是基于最长公共子序列(Longest Common Subsequence, LCS)计算得到的精度和召回率的加权调和平均。一个给定序列的子序列就是该给定序列中去掉零个或者多个元素。给定两个序列X和Y，如果Z既是X的子序列又是Y的子序列，则序列Z是X和Y的一个公共子序列。它不要求词的连续匹配，只要求按词的出现顺序匹配即可，反映句子级的词序。例如对于"ABCD"和"EDCA"，LCS="A"(or "D", "C")；对于"ABCD"和"EACB", LCS="AC"。ROUGE-L的定义为

$$ROUGE-L = \frac{(1+\beta^2)R_{lcs}P_{lcs}}{R_{lcs}+\beta^2 P_{lcs}} \tag{4.9}$$

其中

$$R_{\text{lcs}} = \frac{LCS(X,Y)}{m} \tag{4.10}$$

$$P_{\text{lcs}} = \frac{LCS(X,Y)}{n} \tag{4.11}$$

其中$LCS(X,Y)$是X和Y的最长公共子序列的子序列的长度，m是参考译文的长度，n是机器译文的长度。β在COCO官方评价指标[1]中取1.2。例如对于机器译文“police killed the gunman”，参考译文“police ended the gunman”，最长公共子序列为“police the gunman”，即$LCS(X,Y)$=3，则R_{lcs}=3/4，P_{lcs}=3/4，最终ROUGE-L=0.75。当一条机器译文对应多条参考译文时，将机器译文与参考译文进行逐条比对，取最大的R_{lcs}和P_{lcs}做调和平均计算$ROUGE-L$。

CIDEr[80](Consensus-Based Image Description Evaluation)不同于上述来自机器翻译或机器摘要任务的指标，它是Vedantam在2015年计算机视觉与模式识别大会上提出的针对图片语义描述的基于人们共识的度量标准，与人们的主观判断具有更强的相关性。它通过首先将单词映射到词根，即将“fishes”，

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

“fishing”和“fished”映射为“fish”，然后计算机器描述和参考描述之间的 n-gram 匹配，并且对匹配加以 TF-IDF(term frequency－inverse document frequency)权重得到，其定义为

$$CIDEr(c_i,S_i)=\frac{1}{N}\sum_{n=1}^{N}CIDEr_n(c_i,S) \tag{4.12}$$

n 代表 n-gram，N 在 COCO 官方评价指标中取 4。而 $CIDER_n$ 定义为

$$CIDEr_n(c_i,S_i)=\frac{1}{m}\sum_j\frac{g^n(c_i)\bullet g^n(s_{ij})}{\left\|g^n(c_i)\right\|\left\|g^n(s_{ij})\right\|} \tag{4.13}$$

其中 $S_i=\{s_{i1},s_{i2},...,s_{im}\}$ 是第 i 张图片的机器描述 c_i 对应的 m 条参考描述，$g^n(s_{ij})$ 是参考描述 s_{ij} 的 TF-IDF 向量。即 CIDEr 指标通过计算两个句子的 TF-IDF 向量的余弦距离来度量其相似性。$g^n(s_{ij})$ 定义为 $\left[g_1(s_{ij}),g_2(s_{ij}),...,g_K(s_{ij})\right]$，$g_k(s_{ij})$ 指 s_{ij} 的 n-gram w_k 的 TF-IDF 权重：

$$g_k(s_{ij})=\frac{h_k(s_{ij})}{\sum_{w_l\in\Omega}h_l(s_{ij})}\log\left(\frac{|I|}{\sum_{I_p\in I}\min(1,\sum_q h_k(s_{pq}))}\right) \tag{4.14}$$

$h_k(s_{ij})$ 指 w_k 在 s_{ij} 中的出现次数，$\sum_{w_l\in\Omega}h_l(s_{ij})$ 指 s_{ij} 中n-gram总数，$|I|$ 指图片总数，$\sum_{I_p\in I}\min(1,\sum_q h_k(s_{pq}))$ 指参考描述中出现了 w_k 的图片数目，当 w_k 未出现在参考描述中时取1。通过使用TF-IDF，经常在图片描述中出现的很可能不提供有用信息的n-gram将会被赋予更低的权重，从而使得CIDEr更加符合人们的主观判断。在计算TF-IDF向量的余弦距离时，可以将向量放大n-gram总数倍而不改变其夹角，因而

$$CIDEr_n(c_i,S_i)=\frac{1}{m}\sum_j\frac{f^n(c_i)\bullet f^n(s_{ij})}{\left\|f^n(c_i)\right\|\left\|f^n(s_{ij})\right\|} \tag{4.15}$$

其中

$$f_k(s_{ij})=h_k(s_{ij})\log\left(\frac{|I|}{\sum_{I_p\in I}\min(1,\sum_q h_k(s_{pq}))}\right) \tag{4.16}$$

为了避免一个句子经过人工判断得分很低，但是在自动计算标准中却得分很高的情况，实际应用中采用的CIDEr-D移除了将单词映射到词根的操作，添加了针对机器描述与参考描述长度不一致的情况的高斯惩罚项，并对机器描述的TF-IDF向量进行了截断，即

$$CIDEr-D_n(c_i,S_i)=\frac{10}{m}\sum_j e^{-\frac{(l(c_i)-l(s_{ij}))^2}{2\sigma^2}}\frac{\min(f^n(c_i),f^n(s_{ij}))\bullet f^n(s_{ij})}{\left\|f^n(c_i)\right\|\left\|f^n(s_{ij})\right\|} \tag{4.17}$$

其中 $l(c_i)$ 和 $l(s_{ij})$ 分别代表机器描述和参考描述的长度，σ=6。因子10是为了使CIDEr-D与其它指标处在同一个数量级。

例如对于

图片编号	机器描述	参考描述
1	the gunman murdered police	1.police killed the gunman 2.the gunman was shot down by police
2	on the mat sat the cat	1.the cat sat on the mat 2.there is a cat on the mat

对于 $c_1 = the\ gunman\ murdered\ police$

	$h_k(c_1)$	$h_k(s_{11})$	$h_k(s_{12})$	IDF	$f_k(c_1)$	$f_k(s_{11})$	$f_k(s_{12})$
the	1	1	1	$\log\frac{2}{2}$	0	0	0
gunman	1	1	1	$\log\frac{2}{1}$	$\log 2$	$\log 2$	$\log 2$
murdered	1	0	0	$\log\frac{2}{1}$	$\log 2$	0	0
police	1	1	1	$\log\frac{2}{1}$	$\log 2$	$\log 2$	$\log 2$
killed	0	1	0	$\log\frac{2}{1}$	0	$\log 2$	0
was	0	0	1	$\log\frac{2}{1}$	0	0	$\log 2$
shut	0	0	1	$\log\frac{2}{1}$	0	0	$\log 2$
down	0	0	1	$\log\frac{2}{1}$	0	0	$\log 2$
by	0	0	1	$\log\frac{2}{1}$	0	0	$\log 2$

$CIDEr-D_1(c_1,S_1)=\frac{10}{2}(\frac{2(\ln 2)^2}{\sqrt{3}\ln 2*\sqrt{3}\ln 2}+e^{-\frac{1}{8}}\frac{2(\ln 2)^2}{\sqrt{3}\ln 2*\sqrt{6}\ln 2})=5.413$。同理可得 $CIDEr-D_2(c_1,S_1)$=2.7066992， $CIDEr-D_3(c_1,S_1)$=$CIDEr-D_4(c_1,S_1)$=0，因此 $CIDEr-D(c_1,S_1)$=2.030。同理， $CIDEr-D(c_2,S_2)$=3.7932，因此整个数据集上的得分取每张图片上的得分的平均，为2.9116。

困惑度（Perplexity）用于评价模型对生成的每个单词的确信程度，得分越低越好。被定义为

$$\log_2 PPL(w_{1:L}|I)=-\frac{1}{L}\sum_{n=1}^{L}\log_2 P(w_n|w_{1:n-1},I) \tag{4.18}$$

其中 L 指生成句子的长度，$P(w_n|w_{1:n-1},I)$ 指根据图像 I 和单词序列 $w_{\{1:n-1\}}$，生成下一个单词 w_n 的概率。例如对于“police killed the gunman”，生成每个单词的概率分别是[0.6,0.8,0.3,0.4],那么 $\log_2 PPL(w_{1:L}|I)=0.7135$， $PPL(w_{1:L}|I)$=1.64。

但是上述指标的得分高不一定代表算法根据图像生成的标注语句质量高，和图像内容相符，语法上有没有错误。[81]对 BLEU、ROUGE、METEOR 等评价指标与人工评判结果的相关性进行了研究，结果显示 METEOR 与人类判断的相关性最高，

ROUGE SU-4 和 Smoothed BLEU 其次。而 NIC 模型[17]中，尽管 BLEU-4、METEOR 和 CIDER 上的得分已与人类相近，然而人工评分结果显示，低于 30%的机器描述是完全没有错误的，而绝大多数描述会出现错误，甚至只是与图像有些相关。

4.3 参数调优

实验采用 Intel® Core i7-2600 CPU @3.4GHz 和 32G 内存的 64 位 Ubuntu 16.04 系统，使用了 11G 显存的 NVIDIA GeForce GTX 1080Ti 显卡。所有代码均使用 python 语言编写，基于 TensorFlow 框架构建整体网络结构以及完成训练过程。系统的目标函数是在给定图片集的条件下，出现对应描述的 log 似然函数，即

$$L=\frac{1}{N}\sum_{n}(\sum_{t=1}^{T_n}\log p(w_t^n \mid w_{0:t-1}^n, I^n)+\lambda(1-\sum_{t}^{Tn}\alpha_{ti})) \tag{4.19}$$

其中 I^n 代表数据集中第 n 条描述的对应图片，$w_{0:t-1}^n$ 代表在 w_t^n 之前生成的单词，T_n 是第 n 条描述语句的长度。注意，w_0^n 是统一插到每条描述语句开头的起始符，预测阶段 LSTM 根据起始符生成下一单词，然后将生成的下一单词作为输入，直到生成终止符为止。词向量编码空间大小为 500。优化 LSTM 时，将梯度截断到 5。由于一个批次一般处理多个样本，LSTM 需要执行最长描述长度次迭代，才能计算得到当前批次的损失，并通过反向传播进行一次梯度更新。因此将数据集上的描述的长度截断到 20。

在 coco 上使用不同优化算法损失对模型进行训练，损失下降曲线如图 4.1 所示。使用 Adam 优化器时，损失下降速度明显高于其它算法。由于时间有限，对 Momentum 和 SGD 未做学习率等参数的进一步调优。

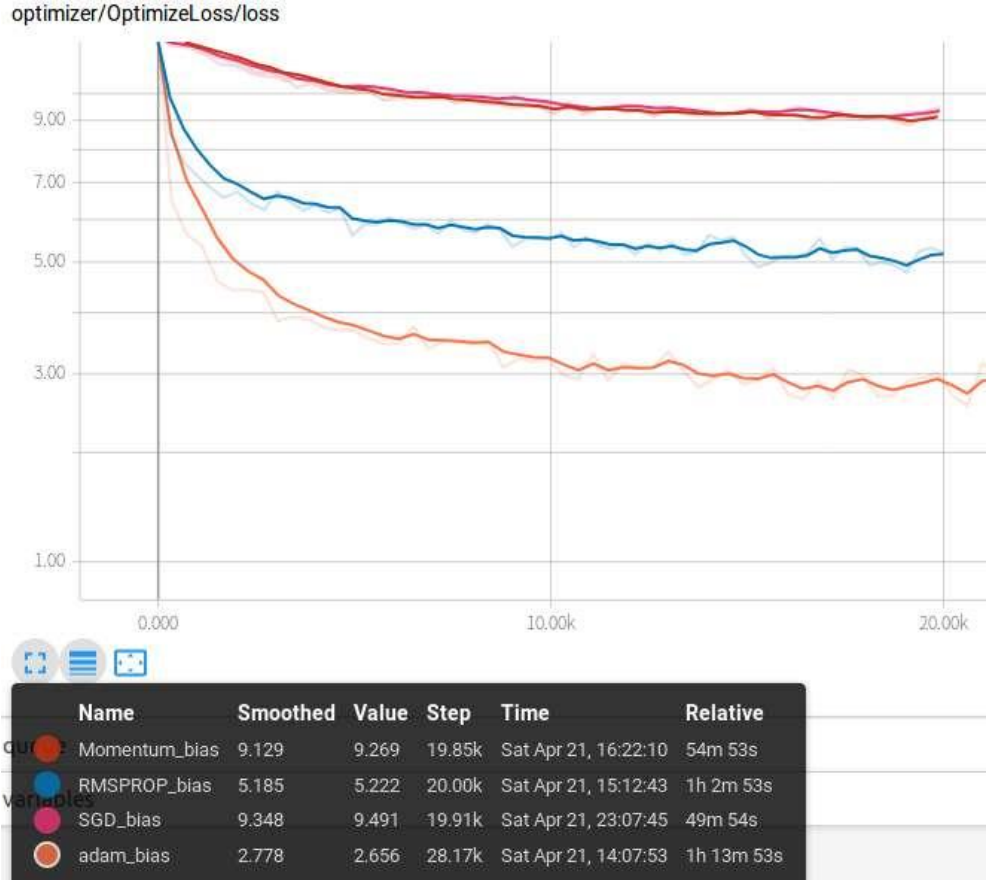


图 4.1 使用不同优化算法进行训练

在 coco 上对使用了不同注意力机制的模型进行优化，损失下降由图 4.2 所示，效果相差不大，采用了根据提取的图片特征和 LSTM 的隐层状态计算偏差量的注意力机制的模型表现略好。

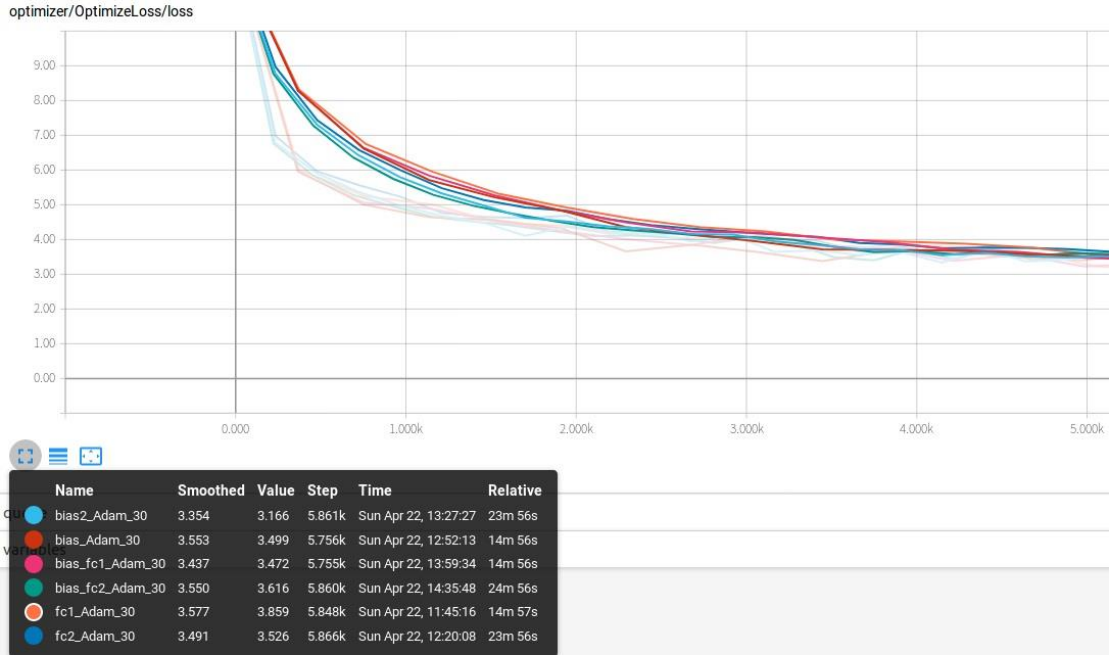


图 4.2 使用不同注意力机制时损失下降曲线

5. 总结与展望

图像描述代表着赋予机器真正理解图像信号的能力，是计算机视觉领域的终极目标。得益于深度神经网络在计算机视觉与自然语言处理领域取得的斐然成果，编解码器架构的提出标志着图像描述算法的研究进入到新的阶段。然而目前虽然能够根据图片生成相应描述，但是词语与图片的相应区域依旧缺乏有效关联，这距离计算机真正能够准确识别图片中的多个目标，并理解目标之间的联系，还有很长的路要走。本文基于区域注意力机制对图片描述任务进行设计和建模，对注意力机制在描述生成过程中的作用进行探索，并实现了web服务器端程序，可以对图片自动生成对应的文字描述。

本文提出了基于区域注意力的图像描述算法。首先使用 Faster R-CNN 模型中的 RPN 生成候选区域，然后通过在最后一层共享卷积层进行 roi pooling 提取图片特征，最后将图像特征通过注意力机制提供给 LSTM 生成描述语句。通过将描述生成过程进行可视化，显示了在描述生成过程中注意力的变化。在此基础上基于 Flask 和 Sqlite 搭建了图像描述生成的服务器端程序，用户可通过浏览器上传图片，查看机器描述并给出评分。管理员可通过后台查看用户上传图片，对机器描述给出的评分，及描述生成过程中的注意力分布，便于查看算法性能，对算法做进一步改进。

不同参数设置对于实验结果会产生很大的影响，如何选择合适的学习率、合

适的词向量编码长度、合适目标检测阈值、合适的 beamsize 等需要在实验中不断的改进。模型损失函数的设计决定着模型能否收敛，对最终生成的图片描述起着决定性作用，也需要在实验中不断调整。除此之外，目前使用 Faster R-CNN 获取图片的区域特征向量与 LSTM 解码生成描述语句是独立的过程，特征编码直接使用了在 COCO 目标检测任务上训练好的权重，未针对图像描述任务做 fine tune。通过将两个模型结合在一起进行训练，应该可以获得损失函数的进一步下降。

参考文献

[1]. He, X. and L. Deng, Deep Learning for Image-to-Text Generation: A Technical Overview. IEEE Signal Processing Magazine, 2017. 34(6): p. 109-116.

[2]. 姜新猛, 基于TensorFlow的卷积神经网络的应用研究, 2017, 华中师范大学.

[3] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg. Baby talk: Understandingand generating image descriptions. In CVPR, 2011.

[4] M. Mitchell, X. Han, J. Dodge, A. Mensch, A. Goyal, A. Berg, K. Yamaguchi, T. Berg, K. Stratos, and H. Daum´e III. Midge: Generating image descriptions from computer vision detections. In EACL, 2012.

[5] A. Gupta and P. Mannem. From image annotation to image description. In ICONIP, 2012.

[6] G. Kulkarni, et al., “Babytalk: Understanding and generating simpleimage descriptions,” IEEE Trans. Pattern Anal. Mach. Intell.,vol. 35, no. 12, pp. 2891–2903, Dec. 2013.

[7] S. Li, G. Kulkarni, T. L. Berg, A. C. Berg, and Y. Choi, “Composingsimple image descriptions using web-scale n-grams,” in Proc. 15thConf. Comput. Natural Language Learn., 2011, pp. 220–228.

[8] Y. Yang, C. L. Teo, H. Daum_e III, and Y. Aloimonos, “Corpusguidedsentence generation of natural images,” in Proc. Conf.Empirical Methods Natural Language Process., 2011, pp. 444–454.

[9] M. Mitchell, et al., “Midge: Generating image descriptions fromcomputer vision detections,” in Proc. 13th Conf. Eur. Chapter Assoc.Comput. Linguistics, 2012, pp. 747–756.

[10] D. Elliott and F. Keller, “Image description using visual dependency representations,” in Proc. Conf. Empirical Methods Natural Language Process., 2013, pp. 1292–1302.

[11] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In NIPS, pages 2121–2129, 2013.

[12] M. Hodosh, P. Young, and J. Hockenmaier. Framing image description as a ranking task: Data, modelsand evaluation metrics. JAIR, 47:853–899, 2013.

[13] R. Socher, Q. Le, C. Manning, and A. Ng. Grounded compositional semantics for finding and describing images with sentences. In TACL, 2014.

[14] P. Kuznetsova, V. Ordonez, A. C. Berg, T. L. Berg, and Y. Choi,“Collective generation of natural image descriptions,” in Proc. 50th Annu. Meeting Assoc. Comput. Linguistics: Long Papers—Vol. 1,2012, pp. 359–368.

[15] P. Kuznetsova, V. Ordonez, T. L. Berg, and Y. Choi, “TREETALK: Composition and compression of trees for image descriptions,”Trans. Assoc. Comput. Linguistics, vol. 2, no. 10, pp. 351–362, 2014.

[16] J. Mao, W. Xu, Y. Yang, J. Wang, and A. Yuille. Deep Captioning with Multimodal Recurrent Neural Networks (m-RNN). In Proc. Int. Conf. Learn. Representations, 2015. 1,2, 4, 5, 6, 7

[17] O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2014. 1, 2, 3, 4, 5, 6, 7
[18] A. Karpathy and L. Fei-Fei. Deep visual-semantic alignments for generating image descriptions. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., 2015. 2, 4, 5, 6, 7
[19] K. Xu, J. Ba, R. Kiros, A. Courville, R. Salakhutdinov, R. Zemel, and Y. Bengio. Show, Attend and Tell: Neural Image Caption Generation with Visual Attention. In Proc. Int. Conf. Mach. Learn., 2015. 2, 5, 6
[20] H. Fang, et al., "From captions to visual concepts and back," in Proc. IEEE Comput. Vis. Pattern Recognit., 2015, pp. 1473–1482.
[21] Q. Wu, C. Shen, L. Liu, A. Dick, and A. van den Hengel, "Whatvalue do explicit high level concepts have in vision to language problems?" in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016,pp. 203–212.
[22] J. Pont-Tuset, P. Arbel´aez, J. Barron, F. Marques, and J. Malik. Multiscale Combinatorial Grouping for Image Segmentation and Object Proposal Generation. InarXiv:1503.00848, March 2015.
[23] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. TPAMI, 2015. 5, 7, 8
[24] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. IJCV, 2013.
[25] R. Girshick, "Fast R-CNN," in IEEE International Conference on Computer Vision (ICCV), 2015.
[26] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In NIPS, 2015.
[27] Region-based Fully Convolutional Networks. In arXiv: 1605.06409, May 2016.
[28] Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-timeobject detection. In: CVPR. (2016)
[29] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed. SSD: Single shot multibox detector. arXiv:1512.02325v2, 2015.
[30] Nair,V. and G.E. Hinton. Rectified linear units improve Restricted Boltzmann machines. ICML 2010.
[31] He, K., et al. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. ICCV 2015.
[32] Djork-Arné Clevert, T.U., Sepp Hochreiter. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). ICLR 2016.
[33] Klambauer G, Unterthiner T, Mayr A, et al. Self-Normalizing Neural Networks[J]. 2017.
[34] Swish: a Self-Gated Activation Function. Prajit Ramachandran, Barret Zoph, Quoc V. Leoc V. Le. arXiv:1710.05941
[35] Glorot X, Bengio Y. Understanding the difficulty of training deep feedforward

neural networks[J]. Journal of Machine Learning Research, 2010, 9:249-256.
[36] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. InProceedings of The 32nd International Conference on MachineLearning, pages 448－456, 2015.
[37] Sebastian Ruder. An overview of gradient descent optimizationalgorithms. arXiv:1609.04747, 2017.
[38] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard,W. Hubbard, and L. D. Jackel, Backpropagation applied tohandwritten zip code recognition. Neural computation, 1989.
[39] Ning Qian. On the momentum term in gradient descent learning algorithms.Neural networks: the official journal of the International Neural Network Society, 12(1):145－151, 1999.
[40]Timothy Dozat. Incorporating Nesterov Momentum into Adam.ICLR Workshop, (1):2013－2016,2016.
[41]Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. International Conference on Learning Representations, pages 1－13, 2015.
[42] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner.Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278－2324,1998.
[43] A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neuralnetworks. In Advances in Neural Information Processing Systems 25, pages 1106－1114, 2012.
[44] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma,Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg,and L. Fei-Fei, “ImageNet Large Scale Visual Recognition Challenge,” in International Journal of Computer Vision (IJCV),2015.
[45] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.
[46] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed,D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich.Going deeper with convolutions. In Proceedings of the IEEEConference on Computer Vision and Pattern Recognition,pages 1－9, 2015.
[47] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna. Rethinking the inception architecture for computer vision.arXiv preprint arXiv:1512.00567, 2015.
[48] Szegedy, C., S. Ioffe and V. Vanhoucke, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. arXiv: 1602.07261, 2016.
[49] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. arXiv:1512.03385, 2015.
[50] Zoph, B., et al., Learning Transferable Architectures for Scalable Image Recognition. arXiv: 1707.07012, 2017.
[51] John J. Hopfeld. Neural networks and physical systems with emergent collective computational abilities. Proceedings of the National Academy of Sciences, 79 (8):2554-2558, 1982.

[52] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," in IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, Nov 1997.
[53] Paul J Werbos. Backpropagation through time: what it does and how to do it.Proceedings of theIEEE, 78(10): 1550 - 1560, 1990.
[54] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory.Neural computation, 9(8): 1735 - 1780, 1997.
[55] Cho, K., et al., Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation. arXiv: 1406.1078, 2014.
[56] Mnih, V., et al., Recurrent Models of Visual Attention. arXiv: 1406.6247, 2014.
[57] Mnih V, Heess N, Graves A. Recurrent models of visual attention[C]. Advances in Neural Information Processing Systems. 2014: 2204-2212.
[58] Bahdanau, D., Cho, K. & Bengio, Y. Neural Machine Translation by Jointly Learning to Align and Translate. Iclr 2015 1 - 15 (2014).
[59] Luong, M. & Manning, C. D. Effective Approaches to Attention-based Neural Machine Translation. 1412 - 1421 (2015).
[60] Rush, A. M. & Weston, J. A Neural Attention Model for Abstractive Sentence Summarization. EMNLP (2015).
[61] Allamanis, M., Peng, H. & Sutton, C. A Convolutional Attention Network for Extreme Summarization of Source Code. 2016.
[62] Tan M, Santos C D, Xiang B, et al. LSTM-based Deep Learning Models for Non-factoid Answer Selection[J]. Computer Science, 2015.
[63] Feng M, Xiang B, Glass M R, et al. Applying deep learning to answer selection: A study and an open task[C]. Automatic Speech Recognition and Understanding. IEEE, 2016:813-820.
[64] Yang, Z. et al. Hierarchical Attention Networks for Document Classification. Naacl (2016).
[65] Wang, L., Cao, Z., De Melo, G. & Liu, Z. Relation Classification via Multi-Level Attention CNNs. Acl 1298 - 1307 (2016).
[66] Zhou, P. et al. Attention-Based Bidirectional Long Short-Term Memory Networks for Relation Classification. Proc. 54th Annu. Meet. Assoc. Comput. Linguist. (Volume 2 Short Pap. 207 - 212 (2016).
[67] Gregor K, Danihelka I, Graves A, et al. DRAW: a recurrent neural network for image generation[J]. Computer Science, 2015:1462-1471.
[68] Ba J, Mnih V, Kavukcuoglu K. Multiple Object Recognition with Visual Attention[J]. Computer Science, 2014.
[69] Mnih V, Heess N, Graves A. Recurrent models of visual attention[C]. Advances in Neural Information Processing Systems. 2014: 2204-2212.
[70] Chen, L., et al., SCA-CNN: Spatial and Channel-wise Attention in Convolutional Networks for Image Captioning. 2016.
[71] Lu, J., et al., Knowing When to Look: Adaptive Attention via A Visual Sentinel for Image Captioning. 2016.

[72] Fu, K., et al., Aligning Where to See and What to Tell: Image Captioning with Region-Based Attention and Scene-Specific Contexts. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017. 39(12): p. 2321-2334.
[73]. Wang, Y. and H. Xiong, Neural image caption generation with global feature based attention scheme[C]. International Conference on Image and Graphics, 2017: Shanghai, China. p. 51 - 61.
[74] Mikolov T, Chen K, Corrado G, et al. Efficient Estimation of Word Representations in Vector Space[J]. Computer Science, 2013.
[75] Mikolov T, Sutskever I, Chen K, Corrado G. and Dean, J. Distributed representations of words and phrases and their compositionality. In: Conference on Advances in Neural Information Processing Systems. Distributed Representations of Words and Phrases and Their Compositionality. 2013. 3111-3119.
[76] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan,P. Doll´ar, and C. L. Zitnick, “Microsoft COCO: Common Objects in Context,” in European Conference on Computer Vision (ECCV), 2014.
[77] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in ACL, 2002.
[78] Banerjee, S. and Lavie, A. (2005) ″METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments″ in Proceedings of Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association of Computational Linguistics (ACL-2005), Ann Arbor, Michigan, June 2005
[79] Lin, Chin-Yew. 2004. ROUGE: a Package for Automatic Evaluation of Summaries. In Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004), Barcelona, Spain, July 25 - 26, 2004.
[80] Vedantam R, Zitnick C L, Parikh D. CIDEr: Consensus-based image description evaluation[C]. Computer Vision and Pattern Recognition. IEEE, 2015:4566-4575.
[81] Elliott D, Keller F. Comparing Automatic Evaluation Measures for Image Description[C]. Meeting of the Association for Computational Linguistics. 2014:452-457.