西北工業大學

# 本科毕业设计论文

**题 目** 基于场景图的图像文字描述

专业名称 计算机科学与技术

学生姓名 衡琪

指导教师 王鹏

毕业时间 2019.6

# 摘 要

序列到序列学习技术及其多种变体在许多任务中都表现出色。现有模型通常只能处理欧几里得空间的规则数据，即使当数据本身是非规则结构化数据时，为了能让现有模型适应处理，只能将结构化数据转换成模型能处理的规则数据，虽然能达成目标，但是在转换过程中很明显地损失了结构化信息，这样就会导致对原始数据信息的不充分利用。为了很好的应对现实中大量存在的结构化非规则数据，大量的研究已经拓展到非欧几里得域，其中最为典型的就是图神经网络及其相关变种，本设计旨在利用图神经网络扩展到序列解码任务。许多机器学习任务的输入能自然地表示为图，而现有的序列到序列模型在实现从图到对应序列的精确转换方面面临重大挑战。为了应对这一挑战，本工作引入了一种通用的端到端的基于图到序列的神经编码器-解码器模型，该结构将输入图映射成节点及图嵌入，并使用基于注意力的长短时记忆力模型对这些矢量进行解码得到目标序列。本工作的方法首先使用改进的基于图的神经网络生成节点和图嵌入，该神经网络具有新颖的聚合策略，以在节点嵌入中结合近邻节点信息。且进一步引入了一种注意力机制，它将节点嵌入和解码序列对齐，以更好地处理大图并且使模型具有更好的解释性。

**关键字：**图，图到序列，注意力，长短时记忆力，嵌入，编码器-解码器

# ABSTRACT

The celebrated Sequence to Sequence learning (Seq2Seq) technique and its numerous variants achieve excellent performance on many tasks. Existing models can only process rule data of Euclidean space. Even when the data itself is irregular structured data, in order to adapt the existing model to processing, only structured data can be converted into rule data that the model can process. Although the goal can be achieved, the structured information is obviously lost during the conversion process, which leads to the underutilization of the original data information. In order to deal well with the large amount of structured irregular data in reality, a large amount of research has been extended to the non-Euclidean domain, the most typical of which is the graph neural network and its related variants. The design is to use the graph neural network. Expand to the sequence decoding task. However, many machine learning tasks have inputs naturally represented as graphs; existing Seq2Seq models face a significant challenge in achieving accurate conversion from graph form to the appropriate sequence. To address this challenge, we introduce a general end-to-end graph-to-sequence neural encoder-decoder architecture that maps an input graph to a sequence of vectors and uses an attention-based LSTM method to decode the target sequence from these vectors. Our method first generates the node and graph embeddings using an improved graph-based neural network with a novel aggregation strategy to incorporate neighbor node information in the node embeddings. We further introduce an attention mechanism that aligns node embeddings and the decoding sequence to better cope with large graphs and makes the model better interpreted.

**KEY WORDS**：graph，graph-to-sequence，attention，LSTM，encoder-decoder

# 目录

# 第一章 绪论

## 1.1 研究背景及意义

现有的神经网络适用的数据大都是规则的欧几里得域数据，例如图像、文本和语音等数据，但现实中存在的大多数据实际上是非规则、结构化的图数据，为了避免这些非规则、结构化数据在转换成规则数据过程中造成信息损失，本设计基于图神经网络直接处理图数据，这对类似于社交网络、化学分子式分析和广告推荐等实际应用具有重大现实意义。

## 1.2 国内外研究现状

### 1.2.1 图像文字描述

图像文字描述任务是给定一幅图像，目标是生成这幅图像对应的文字描述。使用机器正确生成的句子来自动描述图像内容是一项非常具有挑战性的任务，但对此任务的探索和研究可以为社会产生十分有益的影响，例如通过帮助视障人士更好地理解网络上的图像内容、为购物网站种类繁多的物品自动生成合适和吸引消费者的文字描述等。文字描述任务比流行的图像分类或物体检测识别任务挑战性要大得多，这些任务一直是计算机视觉领域的核心任务。实际上，图像描述不仅必须识别图像中包含的对象，还必须关注这些对象之间怎么相互关联以及它们的属性和它们所涉及的动作。

此外，上述图像文本描述必须使用自然语言表达，所以图像描述不仅需要除了视觉理解之外还需要语言模型，进一步加大了整个模型的复杂度。图像文字描述任务通常使用编码器-解码器结构，编码器一般使用卷积神经网络（Convolutional Neural Network）[1]，解码器一般使用循环神经网络（Recurrent Neural Network）[2]，编码器读取图像使用卷积操作得到特征图，解码器读取编码器输出的特征图作为初始化隐状态输入，在每个时间步输出一个单词，直至输出句子终结符。

### 1.2.2 基于场景图的图像文字描述

场景图（Scene Graph，如图 1-1 所示）[3]是把图像内容提取成图表示，使用节点代表原图像中的对象以及对象之间的联系和属性。

基于场景图的图像文字描述，其实是将源输入从像素图像转换成了场景图，场景图显式地表征了图像中不同对象以及对象之间的关系和属性。源输入格式的转变导致处理场景图的网络结构与基于图像的文字描述的网络结构在编码器端

有明显的差别，基于图像的文字描述网络结构以卷积神经网络作为编码工具，而基于场景图的文字描述无法继续使用卷积操作来提取特征，因为卷积神经网络只能处理规则的欧几里得空间的数据，例如图像，这类数据每个像素点周围的近邻像素点的个数是一定的，并且近邻像素点之间的连接关系一致，这种规则排列能很好的使用卷积算子来聚合图像特征，并且可以捕捉数据中存在的不同粒度的信息，而场景图是一类非规则、非欧几里得空间域的复杂数据，不但每个节点周围的近邻节点数目不同，而且节点间的连接关系也复杂多变，现有的卷积算子无法处理这类数据，此项目欲使用最近流行的图神经网络作为编码结构，以场景图作为源输入提取节点级和图级特征，解码器仍然使用典型的循环神经网络用于句子描述的生成。

### 1.2.3 基于非规则图数据的学习理论

卷积神经网络能够利用图像数据的位移不变性、局部连通性和组合性，因此，卷积神经网络可以提取与整个数据集共享的局部有意义的特征，用于各种图像分析任务。

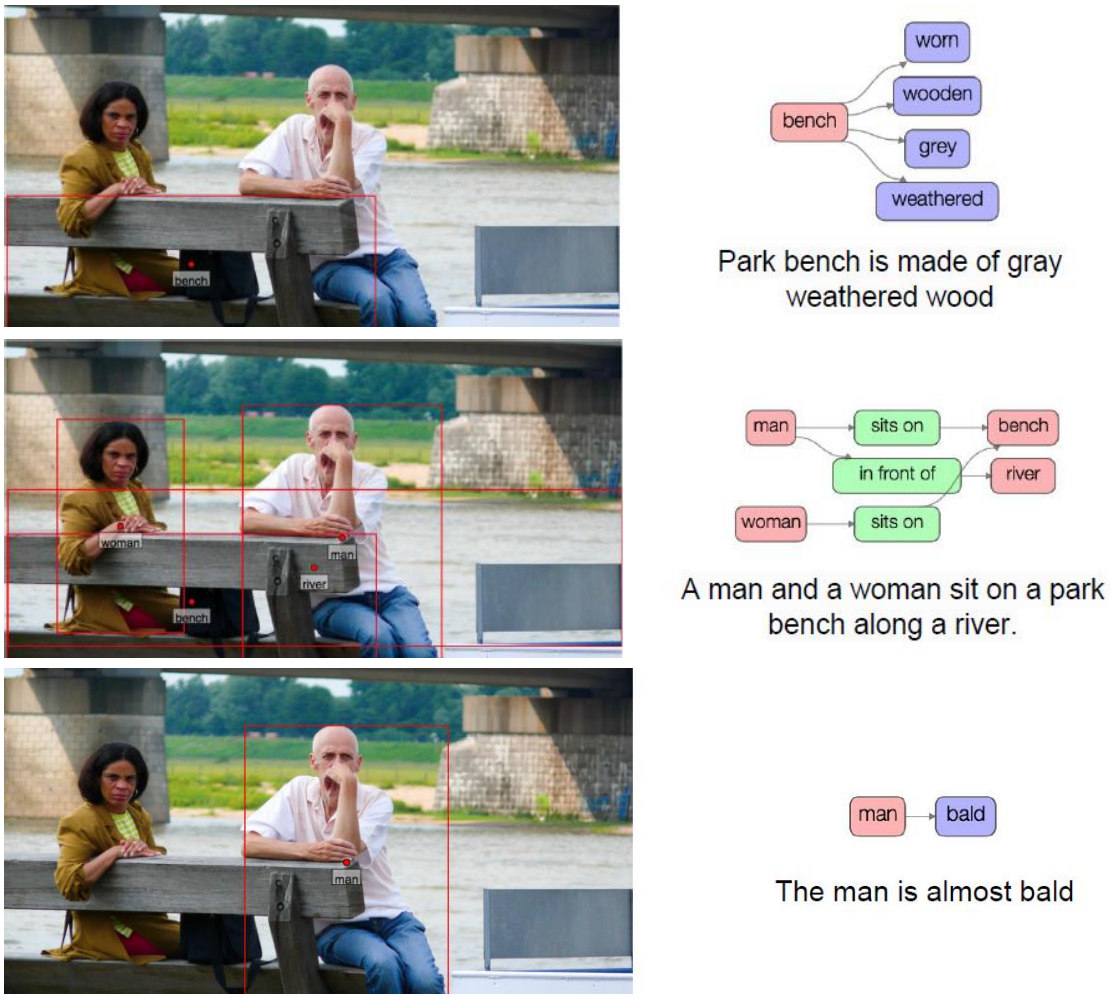
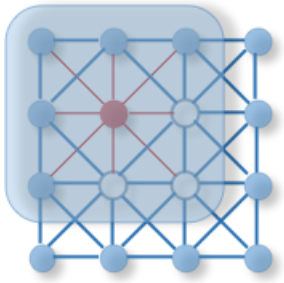


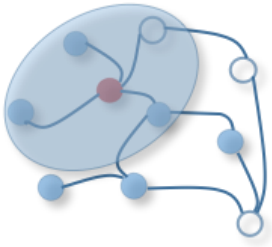

图 1-1 图像到场景图

虽然深度学习在欧几里得数据方面取得了巨大成功，但是越来越多的应用程序从非欧几里得域生成数据并需要进行有效的分析。例如，在电子商务领域，基于图的学习系统能够利用用户和产品之间的交互来提出高度准确的建议；在化学中，分子被建模为图，并且需要确定它们的生物活性以用于药物发现；在引文网络中，论文通过引用相互关联，需要将它们分为不同的组。图数据的复杂性给现有的机器学习算法带来了重大挑战，这是由图数据的不规则性导致的：每个图都拥有可变个数的无序节点，其中每个节点都有不同数量的近邻节点，导致一些在图像领域中重要的计算操作（例如，卷积）不能直接应用于图域数据了。此外，现有机器学习算法的核心假设是实例间彼此独立，然而，对于图数据而言，其中每个实例（节点）通过一些复杂的连接信息与其他（邻居）相关，这些连接信息用于捕获数据之间的相互依赖性，包括引用、关系和交互。

在深度学习成功的推动下，研究人员借鉴了卷积网络、循环网络和注意力网络的思想来设计图神经网络的架构。为了应对图数据的复杂性，新一代图操作理论得到了迅速发展。例如，图 1-2 说明了一种图卷积是如何受到标准 2D 卷积的启发。



（a）2D 卷积。也可视为图结构数据，图像中的每个像素被视为一个节点，其邻居节点由卷积核大小决定。 2D 卷积采用红色节点及其邻居的像素值的加权平均值，节点的邻居是有序的并且具有固定的大小。



（b）图卷积。 为了得到红色节点的隐层表征，图卷积运算的一个简单解决方案是获取红色节点的及其邻居的节点特征的平均值。 与图像数据不同，图数据节点的邻居是无序的并且大小可变。

图 1-2 2D 卷积 vs. 图卷积

## 1.3 论文主要工作

论文主要探索了图神经网络的应用，并基于编码器-解码器模型结构，创新地提出了图到序列的端到端的编码-解码模型结构，并且基于注意力机制提升模型性能。

## 1.4 论文结构

第一章是对相关任务联系与区别的剖析，并且比较了图像卷积和图卷积的区

别与联系；

第二章剖析了序列学习与图到序列学习的联系与区别，并对本工作所作的贡献进行了总结；

第三章对图到序列模型分模块介绍；

第四章详解了图到序列模型；

第五章对整个实验细节和实现细节进行了阐述；

第六章对整篇文章进行了总结，并提出了未来展望。

# 第二章 相关工作

本工作主要的相关工作有三个方面：图像描述、图像到场景图生成和图表征学习。如图 2-1 所示。

## 2.1 图像描述

图像描述任务，输入是图像，输出对应的文字描述。其基于编码器-解码器结构，且图像属于规则数据。著名的序列学习（Seqence-to-Sequence）技术及其众多变体在诸如神经机器翻译、自然语言生成（Natural Language Generation）和语音识别等许多任务中实现了出色的表现。大多数提出的 Seq2Seq 模型可以看作一系列编码器-解码器，其中编码器以序列的形式读取和编码源输入，并将其编码为固定维度的连续矢量表示，而解码器获取编码的矢量并输出目标序列。尽管序列学习具有灵活性和表现力，但此框架的一个显著缺陷是神经网络只能应用于输入表示为序列的问题。

然而，序列可能是最简单的结构化数据，事实上许多重要问题最好用更复杂的结构表示，比如具有更强能力可以编码数据中复杂关系的图结构。此外，移动机器人的路径规划也可以作为图到序列的问题。在本文中，本工作将这类问题表述为图到序列的问题，其将图结构的数据作为输入并产生序列输出。因此，开发一种适用于非欧几里得数据域的方法是十分有必要的，该方法可以学习从图输入到序列输出的映射。

为了应对图到序列的问题，一种简单而直接的方法是将复杂的结构化图数据直接转换为序列，并将序列模型应用于结果序列。然而，Seq2Seq 模型通常无法在这些问题上表现出良好的性能，部分原因是由于复杂结构化数据在转换为序列的过程中会不可避免地造成重要信息的损失，尤其是当输入数据本身就是图结构时更会如此。比将图结构[13] [14]转换为序列更理想的方法是通过提取诸如源句的短语结构之类的句法信息，将源输入转换成树状结构，然后使用 Tree2Seq 结构来处理树状结构。尽管这些方法在某些类型的问题上取得了较为理想的结果，但是大多数所提出的技术在很大程度上取决于基础应用，并且无法以一般的方式推广到更为广泛的问题类别。

## 2.2 图像到场景图生成

图像到场景图的生成能将图像转换成场景图，能增强文字描述的控制性，且能显式的表示图像中的对象、关系和属性。本工作使用的数据集是斯坦福大学提出的 Visual Genome 数据集。

## 2.3 图表征学习

为了解决这种问题，本工作参考[4] [5] [6]提出了图到序列（Graph-to-Sequence）模型，这是一种基于注意力[7]的新型神经网络架构，用于图到序列学习。图到序列模型遵循传统的编码器-解码器方法，具有两个主要组件，图编码器和序列解码器。图编码器旨在学习图的节点嵌入，然后将它们重新组合成相应的图嵌入。为此，在最近的图表示学习方法的启发下，本工作提出了一种基于图的神经网络，通过聚合图中节点的邻域信息来学习节点的嵌入表征，从而探索每个节点的不同特征，得到最终节点嵌入。此外，进一步设计基于循环神经网络的序列解码器，其将图嵌入作为其初始隐藏状态，并通过学习基于与相应节点和所有先前预测相关联的上下文向量来联合对齐和转换来输出目标预测。


图 2-1 相关工作

# 第三章 图到序列模型

## 3.1 详解编码器-解码器

本工作的网络结构也是编码器-解码器结构，因此，了解编码器-解码器的基本工作流程是必要的。编码器-解码器是近年来十分流行的网络架构，这种思想广泛应用于神经机器翻译、语音识别和图像描述等领域。其包含两个组件，第一个组件为编码器，用于对源输入进行编码，得到上下文信息，用作解码器的初始状态，第二个组件为解码器，以编码器的输出为初始状态或者初始输入，根据不同的任务做不同的工作。

为了更好的理解本工作的图到序列模型，首先介绍与图到序列模型基于相似原理的序列模型。序列模型是最典型的编码器-解码器结构之一，例如，序列模型用于支持谷歌翻译、语音设备和在线聊天机器人等应用程序。一般来说，这些应用程序包括：

- **机器翻译**-来自 Google2016 年的论文显示了序列模型的翻译质量如何“接近或超过所有当前发布的最优结果”，如图 3-1 所示。


图 3-1 机器翻译

- **语音识别**-另一篇 Google 论文比较了语音识别任务中现有的序列模型，如图 3-2 所示。


图 3-2 语音识别

- **视频描述**-2015年的一篇论文展示了seq2seq如何在生成电影描述时表现出出色的性能，如图2-4所示。

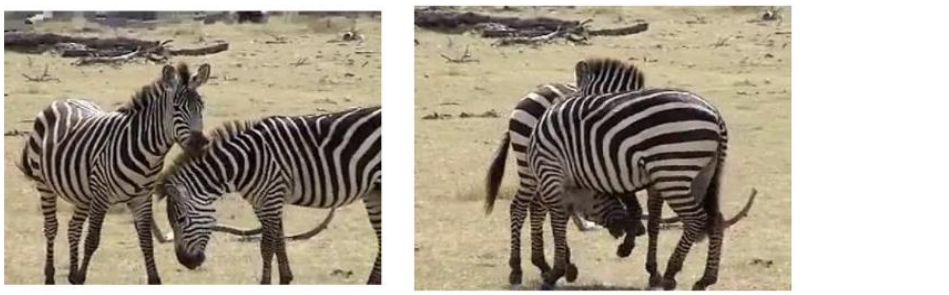



图 3-3 视频描述

这些只是序列模型被视为最佳解决方案的一些应用。该模型可用作任何基于序列的问题的解决方案，尤其是输入和输出具有不同大小和类别的问题。例如，将“你今天做什么？”从中文翻译成英文，输入7个汉字，输出5个单词（What are you doing today?）。显然，我不能使用常规LSTM网络将中文句子中的每个汉字映射到英文句子的单词。这就是序列到序列模型（编码器-解码器）用于解决类似问题的原因，其可以很好的处理输入和输出具有不同大小和类别的问题。

介绍了序列模型的典型应用之后，了解其背后的原理具有十分重要意义。编码器-解码器的典型结构如图5所示，模型结构总共包含三个部分：编码器，中间向量（或称为环境上下文向量）和解码器。

### 3.1.1 编码器

编码器是若干个循环单元的堆叠（长短时记忆单元或门控循环单元会获得更好的性能），其中每个单元接受输入序列的单个元素，收集该元素的信息并向前传播，时刻每个循环神经单元总共需要接收三个输入，分别是$t-1$时刻的细胞状态、$t-1$时刻的隐层状态和$t$时刻当前的输入。

在问答系统中，输入序列是问题中所有单词的集合。 每个单词表示为$x_i$，其中$i$是该单词的顺序。隐藏状态$h_i$使用以下公式计算：

$$h_t = f(W^{(hh)}h_{t-1} + W^{(hx)}x_t)$$

这个简单的公式代表了普通的循环神经网络的结果。我只将适当的权重应用于先前时间步的隐藏状态$h_{t-1}$和输入向量$x_t$。

### 3.1.2 编码向量

- 编码向量是从模型的编码器部分产生的最终隐藏状态。

- 该向量旨在封装所有输入元素的信息，以帮助解码器进行准确的预测。
- 它充当模型的解码器部分的初始隐藏状态。

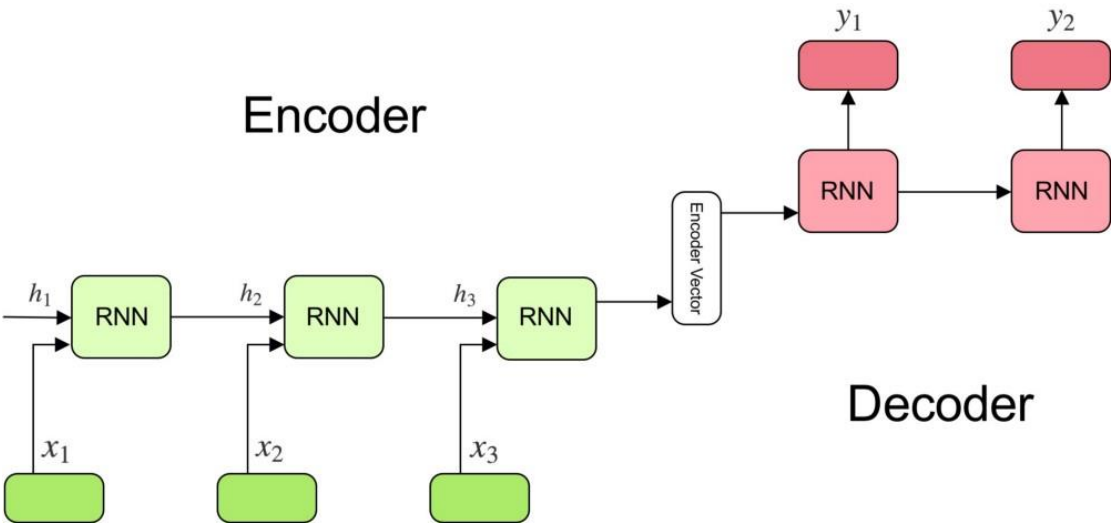



图 3-4 序列学习的编码器-解码器结构

### 3.1.3 解码器

- 若干个循环单元（长短时记忆单元或者门控循环单元）的堆叠，每个循环单元在时间步$t$预测输出$y_t$。每个循环单元接受来$t-1$时刻单元的隐藏状态，并产生和输出以及它自己的隐藏状态。
- 在问答系统中，输出序列是答案中所有单词的集合。 每个单词表示为$y_i$，其中$i$是该单词的顺序。
- 使用以下公式计算任何隐藏状态$h_i$：

$$h_t = f(W^{(hh)} h_{t-1})$$

- 时刻$t$的输出使用如下公式计算：

$$y_t = softmax(W^S h_t)$$

- 使用当前时间步骤的隐藏状态和相应的权重$W^s$来计算输出。$softmax$用于创建概率向量，这将帮助确定最终输出（例如问答环节中的单词）。

## 3.2 解码网络

### 3.2.1 循环神经网络

传统的神经网络无法记忆网络先前看到过的东西，这是解决与时序相关问题的一个主要缺点。例如，假设需要对电影中每个点发生的事件进行分类，传统神经网络无法着手处理此种任务。循环神经网络解决了这个问题，它们是带有循环

的网络，允许信息持续存在，如图 3-5。

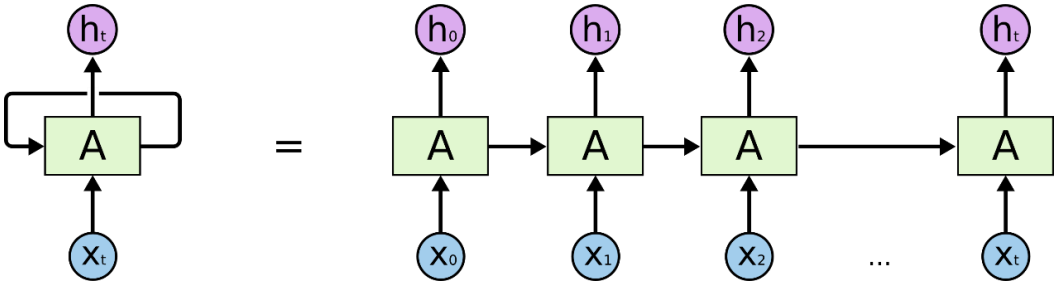



图 3-5 循环神经网络单元的演示

在上图中，神经单元A接收输入$x_t$并输出一个值$h_t$，循环允许信息从网络的一个步骤传递到下一个步骤。可以将循环神经网络视为同一网络的多个副本，每个副本都将消息传递给后继者。理论上，循环神经网络能够处理“长期依赖性”，可以通过仔细挑选参数来解决“长期依赖问题”。但是遗憾的是，在实践中循环神经网络学习这种依赖的能力尚存不足，并且会有梯度爆炸的问题存在，因此，循环神经网络的改进版本长短时记忆网络（LSTM）被探索出来。

### 3.2.2 短时记忆网络

本工作的图到序列模型的解码器组件使用的是 RNN 的改进版本-长短时记忆网络（LSTM），因此，了解 LSTM 的基本理论和工作流程对理解图到序列模型具有十分重要的意义。

长短期记忆网络（Long Short Term Memory Networks，简称为 LSTMs）是一种特殊的 RNN，针对 RNN 不能学习长期依赖和梯度爆炸问题而被提出。所有递归神经网络都具有神经网络重复模块链的形式，在标准 RNN 中，该重复模块具有非常简单的结构，例如单个 tanh 层。LSTM 也具有这种类似链的结构，但重复模块具有不同的结构，下面会逐步介绍 LSTM 细节。

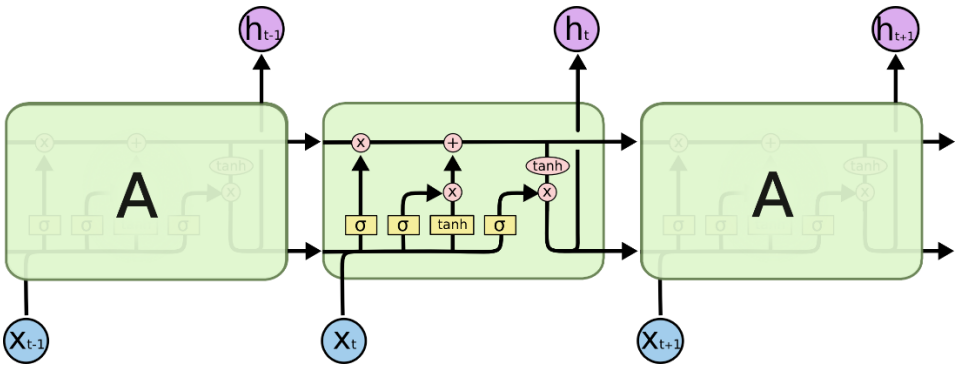



图 3-6 长短记忆网络内部细节展示

LSTM 的关键是细胞状态，水平线贯穿图的顶部。细胞状态有点像传送带，

直接沿着整个链运行，只有一些次要的线性交互，信息很容易沿着它不变地流动，如下图 2-8 所示。

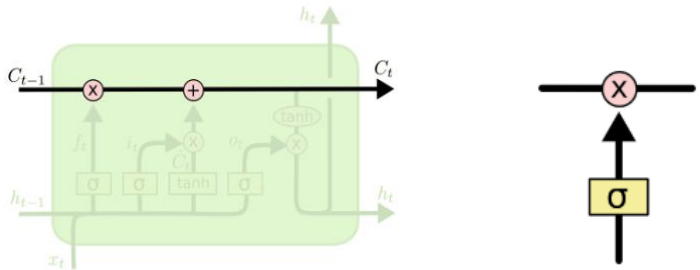
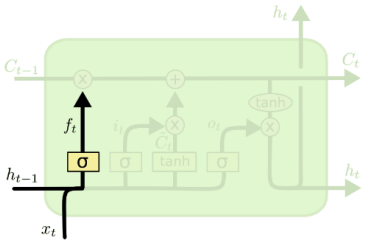



图 3-6 细胞状态

LSTM 确实能够移除或添加信息到细胞状态，这种能力由门控结构掌控，如上图（b）所示。门是一种可选择通过信息的方式，由 sigmoid 神经网络层和逐点乘法运算组成。sigmoid 层输出 0 到 1 之间的数字，描述每个组件应该通过多少信息。值为 0 表示不让任何东西通过，值为 1 表示不限制任何东西。LSTM 具有三个这样的门，用于保护和控制细胞状态。

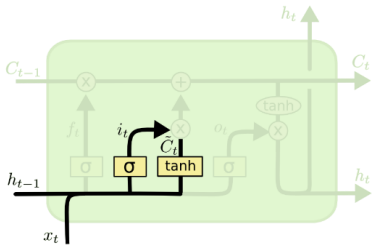
- LSTM 的第一步是确定从细胞状态中丢弃哪些信息，该操作由称为“遗忘门层”的 sigmoid 层决定，它通过查看$h_{t-1}$和$x_t$，为每个细胞状态$C_{t-1}$输出 0 到 1 之间的数字，其中，1 代表“完全保留信息”，而 0 代表“完全丢弃信息”，如图 3-7。




$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

图 3-7 遗忘门

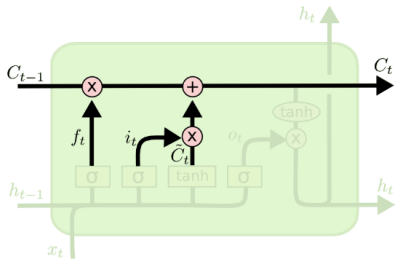
- LSTM 的第二步是决定在细胞状态中存储哪些新信息，这一步有两部分。第一部分是称为“输入门层”的 sigmoid 网络层决定将更新哪些值，接下来，tanh 层创建新的候选值向量$\tilde{C}_t$，此向量可以被加到状态中。在下一步中，将结合这两个向量来更新状态，如图 3-8。




$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

图 3-8 输入门

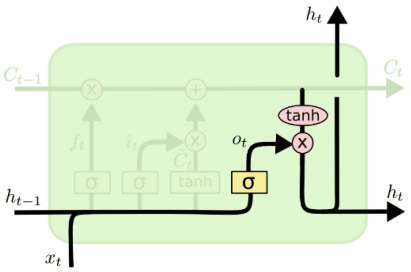
- LSTM 的第三步是将旧的细胞状态$C_{t-1}$更新为新的细胞状态$C_t$。之前的步骤已经决定要做什么，只需要根据这些决定去实践即可。将旧状态乘以$f_t$，表示忘记那些之前决定忘记的事情，然后添加$i_t * \tilde{C}_t$。这是新的候选值向量，决定新的状态应该添加多少新的信息，如图 3-9。




$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

图 3-9 更新门

- LSTM 的最后一步是决定需要输出的内容。此输出基于过滤后的细胞状态。首先，通过一个 sigmoid 层来决定要输出细胞状态的哪些部分。然后，将细胞状态通过 tanh（将值推到介于-1 和 1 之间）层并将其乘以 sigmoid 门的输出，以便只输出决定输出的部分，如图 3-10。




$$o_t = \sigma\left(W_o \left[h_{t-1}, x_t\right] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

图 3-10 更新

### 3.2.3 图神经网络

图神经网络是处理图非规则数据的典型力作，在了解图卷积、图注意力和门控图网络前理解图神经网络的相关知识是十分必要的。

图神经网络（Graph Neural Network, 简称为 GNN）是根据图结构$\mathrm{G} = (V, E)$定义的通用神经网络体系结构。节点$v \in \mathrm{V}$取$1, \ldots, |V|$中的唯一值，边是对$\mathrm{e} := (v, v') \in \mathrm{V} \times \mathrm{V}$。

本段在介绍时将专注于有向图（GNN 可以很容易地适用于无向图当中），所以

$(v,v')$代表有向边$v \to v'$。节点$v$的*节点向量*（或*节点表示*或*节点嵌入*）由$h_v \in R^D$表示，图也可能包含每个节点$v$的节点标签$l_v \in \{1,\dots,L_v\}$和每条边的边标签或边类型$l_e \in \{1,\dots,L_E\}$。接下来将重载符号，当$S$是一组节点时让$h_S := \{h_v | v \in S\}$，并且当$S$是一组边时让$l_S := \{l_e | e \in S\}$。函数$\mathrm{IN}(v) := \{v' | (v',v) \in E\}$返回前导节点$v'$的集合，其中$v' \to v$，类似地，函数$\mathrm{OUT}(v) := \{v' | (v,v') \in E\}$是边$v \to v'$的后继节点$v'$的集合。与$v$相邻的所有节点的集合是$\mathrm{NBR}(v) := \mathrm{IN}(v) \cup \mathrm{OUT}(v)$，并且从$v$传入或从$v$传出的所有边的集合是$\mathrm{CO}(v) := \{(v',v'') \in E | v := v' \forall v := v''\}$。

GNN 通过两个步骤将图映射到输出。首先，有一个传播步骤，计算每个节点的节点表示;第二，输出模型$O_v := g(h_v,l_v)$从节点表示和相应的标签映射到每个$v$的输出$O_v$。在$g$的表示法中，依赖性隐含在参数上，该模型可以端到端训练，因此所有参数都是使用基于梯度优化共同学习的。

1） 传播模型

传播模型是利用迭代过程传播节点表示。初始节点表示$\mathbf{h}_v^{(1)}$被设置为任意值，然后每个节点表示根据如下公式循环更新直到收敛，其中$t$表示时间步长：

$$\mathbf{h}_v^{(t)} = f^*(l_v, l_{\mathrm{CO}(v)}, l_{\mathrm{NBR}(v)}, \mathbf{h}_{\mathrm{NBR}(v)}^{(t-1)}).$$

Scarselli 等人讨论了几种变体，包括位置图式、特定节点更新和邻域替代表示。具体而言，Scarselli 等人建议$f^*(\cdot)$分解为每个边项的总和：

$$f^*(l_v, l_{\mathrm{CO}(v)}, l_{\mathrm{NBR}(v)}, \mathbf{h}_{\mathrm{NBR}(v)}^{(t)}) = \sum_{v' \in \mathrm{IN}(v)} f(l_v, l_{(v',v)}, l_{v'}, \mathbf{h}_{v'}^{(t-1)}) + \sum_{v' \in \mathrm{OUT}(v)} f(l_v, l_{(v,v')}, l_{v'}, \mathbf{h}_{v'}^{(t-1)}),$$

其中$f(\cdot)$是$h_{v'}$的线性函数或神经网络。$f$的参数取决于标签设置，例如在下面的线性情况下，$\boldsymbol{A}$和$\boldsymbol{b}$是可学习的参数：

$$f(l_v, l_{(v',v)}, l_{v'}, \mathbf{h}_{v'}^{(t)}) = \mathbf{A}^{(l_v, l_{(v',v)}, l_{v'})} \mathbf{h}_{v'}^{(t-1)} + \mathbf{b}^{(l_v, l_{(v',v)}, l_{v'})}.$$

2） 输出模型

输出模型是按节点定义且是一个映射到输出的可微函数$g(h_v,l_v)$，其通常是线性或神经网络映射。*Scarselli* 等关注每个节点独立的输出，通过将每个节点$v \in \mathrm{V}$的最终节点表示$h_v^{(T)}$映射到输出$O_v := g\left(h_v^{(T)}, l_v\right)$来实现。为了解决图级分类任务，*Scarselli* 等人建议创建一个虚拟的“超级节点”（super node），其通过特殊类型的边连接到所有其他节点。因此，图级回归或分类任务可以以与节点级回归或分类相同的方式处理。

训练学习是通过 Almeida-Pineda 算法完成的，该算法通过传播更新到收敛，此算法的优点是不需要存储中间状态以便计算梯度，缺点是必须约束参数，以便传播步骤是一个压缩映射，这点是收敛所必需的，但可能会限制模型的性能。当$f(\cdot)$是神经网络时，推荐使用网络雅可比行列式的 1 范数的惩罚项。

本模型从图表示学习、图神经网络和神经编码器-解码器模型的研究领域中获得灵感。

### 3.3 图表示学习

事实证明，图表示学习对于广泛的基于图的分析和预测任务非常有用。图表示学习的主要目标是学习将节点嵌入到低维向量空间中的点的映射。这些表示学习方法可以大致分为两类，包括基于矩阵因子分解的算法和基于随机游走的方法。一系列研究通过矩阵分解来学习图节点的嵌入。这些方法直接训练用于训练和测试数据的各个节点的嵌入。另一系列工作是使用基于随机游走的方法，通过探索单个大尺度图的邻域信息来学习节点的低维嵌入。

图注意力和门控图网络通过使用节点属性结合聚合邻域信息进行归纳学习节点嵌入的技术。本工作的图编码器是图注意力和门控图网络，两者的处理思路十分相似，都是聚合邻域节点信息更新中心节点信息，区别是聚合策略不同，前者基于注意力机制给不同邻域赋予不同的权重进而聚合不同程度的邻域信息，后者基于对称邻接矩阵在时间步上对邻域节点信息进行聚合。其次，本工作利用两种不同的方案基于节点嵌入生成图嵌入，一种是基于所有节点嵌入进行全局池化而得到图嵌入，另一种是基于所有节点嵌入进行平均池化得到图嵌入。

### 3.4 图神经网络

在过去几年中，基于图神经网络（Graph Neural Network）[12]学习图节点或整个（子）图的表示的方法激增，这些图神经网络将包括循环神经网络和卷积神经网络在内的网络架构扩展到图数据域。一系列研究以循环神经网络为基础探索针对图数据的神经网络。在原始图神经网络框架中引入循环神经网络（使用门控循环单元 GRU 更新）进行实践。

门控图神经网络是重要的处理图数据的图神经网络，其需要在训练期间给出完整邻接矩阵算子并且适用于无向图和有向图。它本质上是一个预测模型，学习预测图中的嵌入。门控图神经网络对于图卷积神经网络，类似于循环神经网络对于卷积神经网络。

### 3.5 编码器-解码器模型

最成功的编码器-解码器架构之一是序列学习模型，它们最初被提议用于机器翻译。最近，经典的序列到序列模型及其变体已经应用于几个应用程序，这些模型可以执行从对象到序列的映射，包括从图像到句子的映射，来自 python 程序的问题语句映射到对应的解决方案（程序的答案）。很容易看出，示例中的序列对象通常自然地以图为结构表示而不是序列表示。

### 3.6 数据准备

深度学习尽管在诸如图像分类、目标检测等任务方面取得了进步，但在诸如图像描述和问题回答之类的认知任务上仍然表现不佳。认知是任务的核心，不仅涉及识别，还涉及我的视觉世界。但是，用于处理认知任务图像中丰富内容的模

型仍在使用为感知任务设计的相同数据集进行训练。为了在认知任务中取得进步，模型需要理解图像中对象之间的交互和关系。当被问及“乘坐什么车辆？”时，计算机将需要识别图像中的物体以及骑车（人，马车）和拉动（马，马车）的关系，以便正确回答“人在乘坐马车”。

Visual Genome 数据集是一种可以建模这种关系的数据集，其收集每个图像中对象、属性和关系的注释，以便于模型学习这些关系和属性。具体来说，Visual Genome 数据集包含超过 100K 的图像，其中每个图像平均有 21 个对象，18 个属性和 18 个对象之间的成对关系。其将区域描述中的对象、属性、关系和名词短语规范化。如图 3-11 所示。

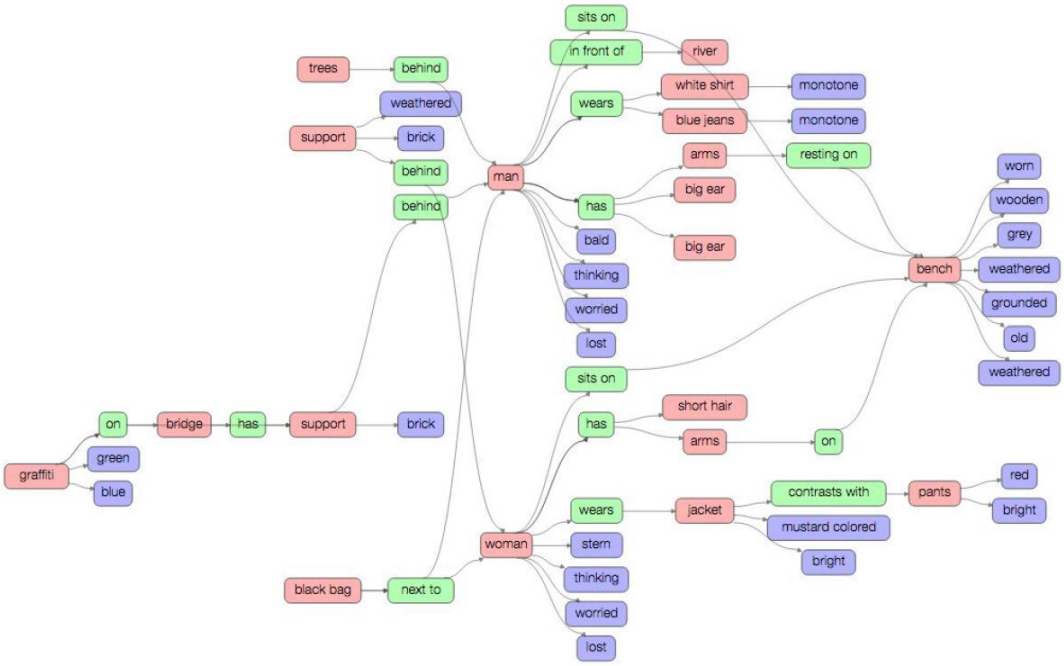



图 3-11 Visaul Genome 数据集可视化演示

# 第四章 实验设置

## 4.1 模型框架

如图 5-1，图-序列模型包括图编码器、序列解码器和节点注意力机制。遵循第四章第一接讲述的编码器-解码器架构，图编码器首先根据图生成节点嵌入，然后基于学习的节点嵌入构造图嵌入。最后，序列解码器将图嵌入和节点嵌入作为输入，并在生成序列时对节点嵌入进行注意力。后面，首先介绍节点嵌入生成算法，该算法通过聚合来自图中节点邻域的信息来生成最终节点嵌入，在这些节点嵌入时，使用了两种方法来生成捕获整个图信息的图嵌入。

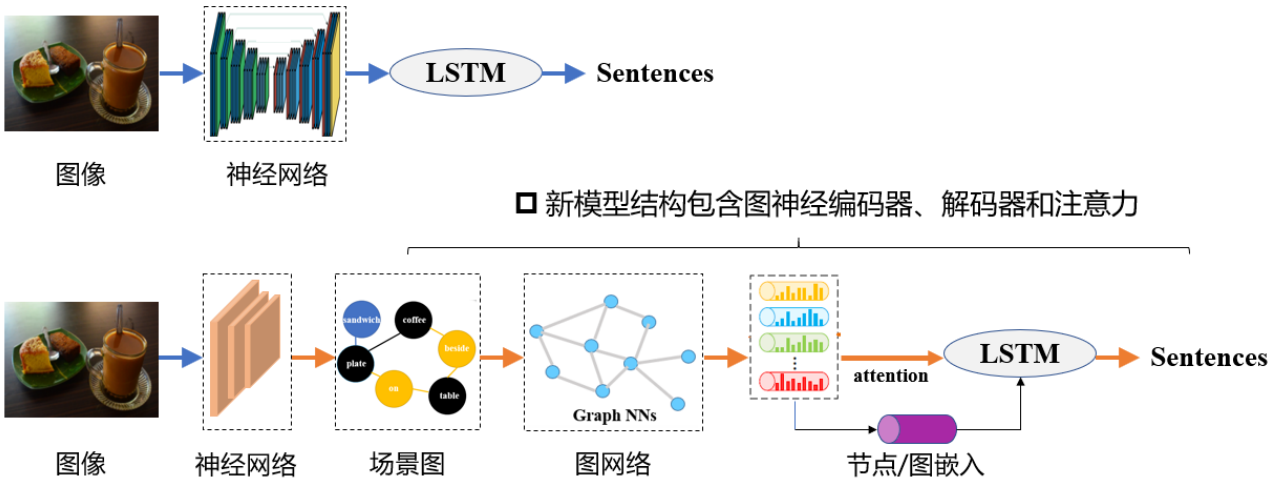



图 4-1 图到序列模型结构

图 4-1（上）展示的是图像描述任务，其输入是一幅图像，经过卷积神经网络，得到图像特征，输入到长短时记忆力网络中进行解码得到句子。图 4-1（下）是场景图描述任务，首先得到图像的场景图，在经过图神经网络更新得到图嵌入和节点嵌入，输入到基于注意力的长短时记忆里网络中进行解码得到句子。

## 4.2 数据处理

**获取三元组。**以 Visual Genome 数据集的 region_graph.json 文件所给数据为基础，其中包含层层字典嵌套，关键的键是“objects”，“phrase”，“attributes”和“relationships”，首先确定以三元组的形式表示图，即〈subject,predict,object〉，为了得到这种三元组，以 objects 键作为基础，以 relationships 和 attributes 键为关系建立三元组，这样就得到了 region_graph 的所有三元组表示，这些三元组组合起来就能得到场景图。其中，如图 5.1 所示，对象、关系和属性都将其表示成

图中单独的节点，以便进行节点更新和节点嵌入表示。

**字典生成。**对于训练集中的所有图和短语，将其使用空格分离，去除标点符号并且全部转换为小写字母，依次统计单词出现频率。之后使用集合去重，对所有单词进行编号。最重要的是添加四个特殊符号：填充<pad>，开始<start>，结束<end>和未知<unk>。

**节点初始嵌入生成。**其中，由于每个图中节点数目是不定的，为了方便程序统一处理一个批量，需要将每个图中节点数目填充至所在批量中图节点数目的最大值，在本工作中使用数值 0 作为填充值。在获得每个批量的填充后的图数据后，根据字典的大小和设置嵌入维度得到嵌入矩阵，其中行数等于字典大小，列数等于嵌入维度，此嵌入矩阵的值都是可学习的参数，根据整个模型结构的误差函数来更新学习每个单词的嵌入值。这样，就得到了每个单词节点的初始嵌入表示。

**邻接矩阵生成。**要想使用图神经网络处理节点嵌入并得到更新后的节点嵌入和图级嵌入，必须得到对应图的邻接矩阵。首先，根据前面生成的三元组，对所有的三元组使用集合操作处理之后得到不重复的节点，其数目等于邻接矩阵的行与列数，所有值初始化为 0，表示没有任何连接边。随后对此字典中的单词编号存储到字典中，键为单词，值为编号值，这样操作之后每个单词都会获得一个唯一的编号，之后遍历三元组，将三元组中的每个单词在编号字典中查看是否存在，如果存在，则记录对应的编号并在邻接矩阵对应的位置置 1，表示对应的两个节点之间有边连接。操作完成之后，得到整个图的邻接矩阵表示。

但这样还没完成，因为如果直接使用这个邻接矩阵去操作更新节点嵌入的话，节点嵌入会失去自身的信息，所以要给此邻接矩阵在加上一个单位矩阵，这样，就得到最终的邻接矩阵表示。

**数据加载。**使用自定义的数据加载器类，自定义类，其可以获得一个样本对，本工作中包含图和真值短语。重写方法,此方法可以获取一个批量的样本，并且在此方法中完成批量填充，对不同大小的图和相应的邻接矩阵进行填充。

### 4.3 节点嵌入生成

此工作的节点嵌入生成主要遵循了门控图神经网络和图注意力网络。

#### 4.3.1 门控图神经网络

门控图神经网络是在门控循环单元（GRU）的基础上改进而来的，所以其对节点嵌入的更新基于时间步，通过在每个时间步使用邻接矩阵对节点特征矩阵进行近邻节点特征聚集，得到最终节点嵌入，公式如下，下面将结合公式进行详细介绍。

$$h_v^{(1)} := [x_v^T, 0]^T \tag{1}$$

$$\mathrm{a}_v^{(t)} := A_v^T \left[h_1^{(t-1)^T} \ldots h_{|V|}^{(t-1)^T}\right]^T + b \tag{2}$$

$$z_v^t := \sigma\left(W^z \mathrm{a}_v^{(t)} + U^z \mathrm{h}_v^{(t-1)}\right) \tag{3}$$

$$r_v^t := \sigma\left(W^r \mathrm{a}_v^{(t)} + U^r \mathrm{h}_v^{(t-1)}\right) \tag{4}$$

$$\tilde{h}_v^{(t)} := tanh\left(W \mathrm{a}_v^{(t)} + U\left(r_v^t \circ \mathrm{h}_v^{(t-1)}\right)\right) \tag{5}$$

$$\mathrm{h}_v^{(t)} := (1 - z_v^t) \circ \mathrm{h}_v^{(t-1)} + z_v^t \circ \tilde{h}_v^{(t)} \tag{6}$$

- 公式（1）是对节点注释填充 0 得到初始节点表示，如果初始输入是不同长度的节点向量的话需要对整个批量的节点向量填充至相同长度，本工作的节点注释相当于初始节点嵌入，本身就是相同长度的节点嵌入，所以无需再填充。
- 公式（2）是在时间步循环内进行的第一步操作，这是门控图神经网络的核心步骤，此步骤使用先前生成的邻接矩阵对每个前一个时间步得到节点特征矩阵进行信息聚集，通过使用邻接矩阵与节点特征矩阵进行矩阵相乘，得到当前时刻的节点特征矩阵；
- 公式（3）相当于更新门，使用更新后的节点特征矩阵与前一个时间步的节点特征矩阵相连接，经过一个简单的神经网络层后降维更新到与节点特征矩阵相同的维度，之后再通过 sigmoid 层进行门控值的控制，将在后面用于决定保留多少当前时刻的特征；
- 公式（4）相当于遗忘门，与更新门类似，使用更新后的节点特征矩阵与前一个时间步的节点特征矩阵相连接，经过一个简单的神经网络层后降维更新到与节点特征矩阵相同的维度，之后再通过 sigmoid 层进行门控值的控制，将在后面用于决定遗忘多少上一个时间步的特征，将与更新门协同使用；
- 公式（5）代表当前时刻的节点特征，由当前时刻的节点特征矩阵与和遗忘门点乘上一时刻节点特征矩阵得到的特征矩阵相连接，经过一个简单的神经网络层后降维更新到与节点特征矩阵相同的维度，之后再通过 tanh 层进行非线性激活，得到当前时刻的最终节点特征；
- 公式（6）决定保留多少当前时刻的特征，遗忘多少上一个时间步的特征。

- 重复步骤（2）~（6）若干次。

经过上述更新$T$个时间步之后，便可以得到聚集了近邻节点信息的节点特征矩阵。

### 4.3.2 图注意力网络

首先描述单个图注意力层，其作为实验中所有图注意力网络（Graph Attention Network, 简称为 GAT）架构中使用的唯一层。图层的输入是一组节点特征向量$\mathrm{h}:=\{\overrightarrow{h_1},\overrightarrow{h_2},\ldots,\overrightarrow{h_N}\},\overrightarrow{h_i}\in R^F$，其中 N 是节点数，F 是每个节点中的特征数。该层产生一组新的节点特征$h':=\left\{\overrightarrow{h_1'},\overrightarrow{h_2'},\ldots,\overrightarrow{h_N'}\right\},\overrightarrow{h_i'}\in R^{F'}$作为其输出。

为了获得足够的表达能力将输入特征转换为更高级别的特征，至少需要一个可学习的线性变换。为此，作为初始步骤，通过权重矩阵$\mathrm{W}\in R^{F'\times F}$进行参数化的共享线性变换被应用于每个节点。然后，对节点执行自注意力（SelfAttention）操作--一个共享的注意力机制$R^{F'}\times R^{F'}\to R$用来计算注意力系数：

$$e_{ij}=a(\mathbf{W}\vec{h}_i,\mathbf{W}\vec{h}_j) \tag{1}$$

注意力系数代表了节点$j$的特征对节点$i$的重要性。在注意力一般的表述意义中，其允许丢弃所有结构信息让每个节点参与其他所有节点的更新。通过执行屏蔽注意力将图结构注入到机制中-对节点$j\in N_i$计算系数$e_{ij}$，其中$N_i$是途中节点$i$是邻域节点，在本实验中，所有邻域将是$i$（包括$i$）的一阶邻域。为了使系数在不同节点之间易于比较，使用 $softmax$ 函数在 $j$ 的所有选项中对它们进行标准化：

$$\alpha_{ij}=\mathrm{softmax}_j(e_{ij})=\frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e_{ik})}. \tag{2}$$

在实验中，注意力机制$a$是一个单层反馈神经网络，使用权重向量$\vec{\mathrm{a}}\in R^{2F'}$参数化，并且使用 LeakyReLU 非线性进行激活。因此，使用注意力机制计算的系数（如图 5.2 左所示）可以表达为：

$$\alpha_{ij}=\frac{\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathrm{a}}^T[\mathbf{W}\vec{h}_i\|\mathbf{W}\vec{h}_j]\right)\right)}{\sum_{k\in\mathcal{N}_i}\exp\left(\mathrm{LeakyReLU}\left(\vec{\mathrm{a}}^T[\mathbf{W}\vec{h}_i\|\mathbf{W}\vec{h}_k]\right)\right)} \tag{3}$$

其中，$\cdot^T$代表转置，$||$代表连接。

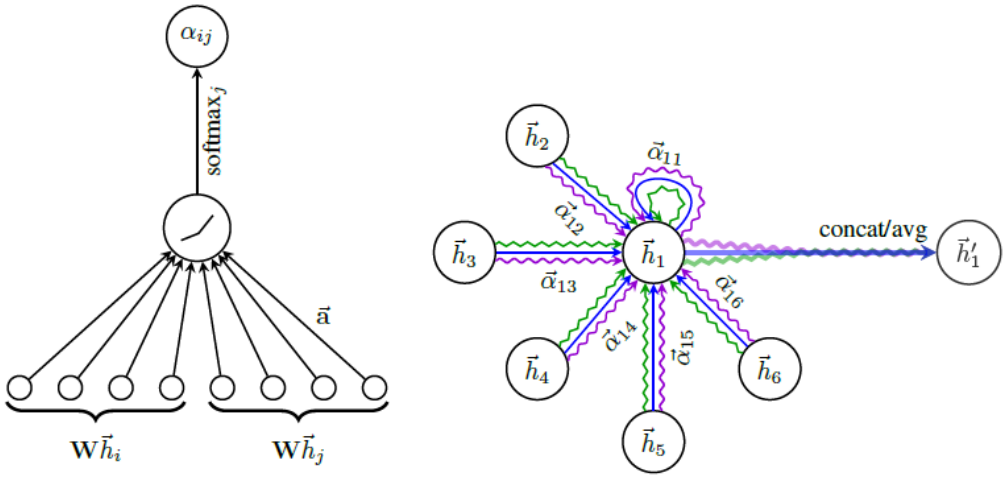



图 4-2

图 4-2 左：在本模型中使用的注意力机制，由权重向量参数化，并且使用 LeakyReLU 非线性进行激活。右：节点 1 邻域的多头注意力（多头数目 K=3），不同的箭头类型和颜色表示不同的注意力计算。从不同头聚集的特征通过连接或者平均来获得新状态。

一旦获得归一化的注意力系数，其会和对应的节点特征向量做线性结合，作为每个节点最终的输出特征（可能使用非线性激活，$\delta$）：

$$\vec{h}'_i = \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}\mathbf{W}\vec{h}_j\right). \tag{4}$$

为了稳定化自注意力的学习过程，拓展使用了多头机制来使自注意力更加有效。特别的，$K$个独立的注意力机制执行了公式（4）的变换，之后将得到的特征连接，因此，最后的输出特征表示为：

$$\vec{h}'_i = \Big\Vert_{k=1}^{K} \sigma\left(\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k\vec{h}_j\right) \tag{5}$$

公式中，||代表连接，$a_{ij}^k$是由第$k$个多头注意力机制计算的注意系数，$W^k$是对应的线性变换权重矩阵。特别的，如果在最终预测层，连接就没必要，相应的应该执行平均化：

$$\vec{h}'_i = \sigma\left(\frac{1}{K}\sum_{k=1}^{K}\sum_{j\in\mathcal{N}_i} \alpha_{ij}^k \mathbf{W}^k\vec{h}_j\right) \tag{6}$$

对应本工作而言，通过连接可以获得最终的节点嵌入，而平均池化可以作为图级嵌入作为输出。

### 4.4 图级嵌入生成

图级嵌入能针对整个图基于节点嵌入获得全局信息，图神经网络的大多数现有工作更多地关注节点嵌入而不是图嵌入，因为它们关注于节点分类任务。然而，

传达整个图信息的图嵌入对于下游解码器是必不可少的。在本工作中，引入了两种方法（即，基于平均池化和基于最大池化）来从节点嵌入生成这些图嵌入。

基于平均池的图嵌入（如图 4-3）。在这种方法中，基于所有节点特征向量，在每个维度上对不同节点的值求平均，在直觉上，能在每个特征处求得所有节点的平均表示，将此向量作为全局图表示。

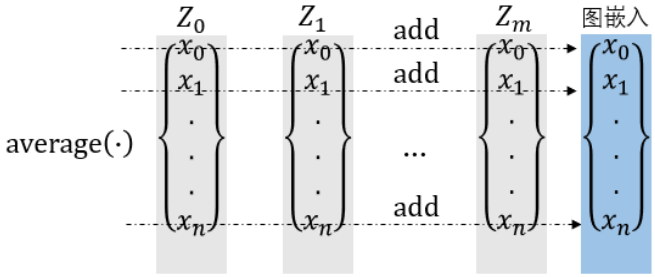



图 4-3 基于平均池的图嵌入

基于最大池的图嵌入（如图 4-4）。这种方法同样基于所有节点特征向量，在每个维度对所有节点的值求最大值，此方法能在每个特征处求得主特征，此向量作为全局图表示。

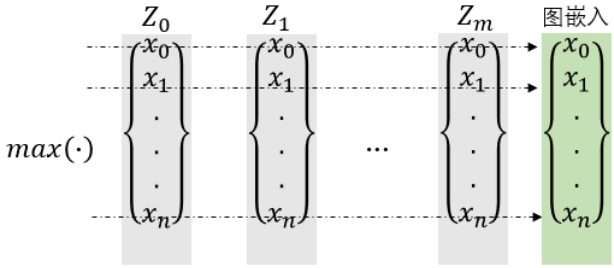



图 4-4 基于最大池的图嵌入

本工作采用基于平均池的图嵌入，因为从效果上比较，基于平均池化的全局图表示略胜一筹。

## 4.5 基于注意力的解码器

### 4.5.1 注意力

序列解码器是循环神经网络（RNN）的改进版本长短时记忆网络（LSTM），给定所有先前的词$y_{<i} := y_1, \cdots, y_{i-1}$，时间步$i$的循环神经网络的隐状态$s_i$和能将注意力引向编码器特征向量的上下文向量$c_i$，循环神经网络根据这些预测下一个单词$y_i$。特别地，上下文向量$c_i$取决于由图编码器将输入图映射到的一组节点表示$(z_1, z_2, \ldots, z_V)$。每个节点表示$z_i$包含关于整个图的信息，其强烈关注于输入图的第$i$个节点周围的部分。上下文向量$c_i$基于这些节点表示计算加权和，计算每个节点表示的权重$a_{ij}$：

$$c_i = \sum_{j=1}^{\mathcal{V}} \alpha_{ij} h_j, \ where \ \alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{\mathcal{V}} \exp(e_{ik})}, \ e_{ij} = a(s_{i-1}, h_j) \tag{2}$$

其中$a$是一个对齐模型，用于评估位置$j$周围的输入节点和位置$i$的输出匹配的程度。得分基于循环神经网络的隐状态$s_{i-1}$和输入图的第$j$个节点表示。将对齐模型$a$参数化为前馈神经网络，该网络与所提出的系统的其他组件共同训练。模型经过联合训练，以在给定源图的情况下最大化正确描述的条件对数概率。在推理阶段，使用贪婪算法得到描述。

注意力机制具体细节如图 4-5 所示

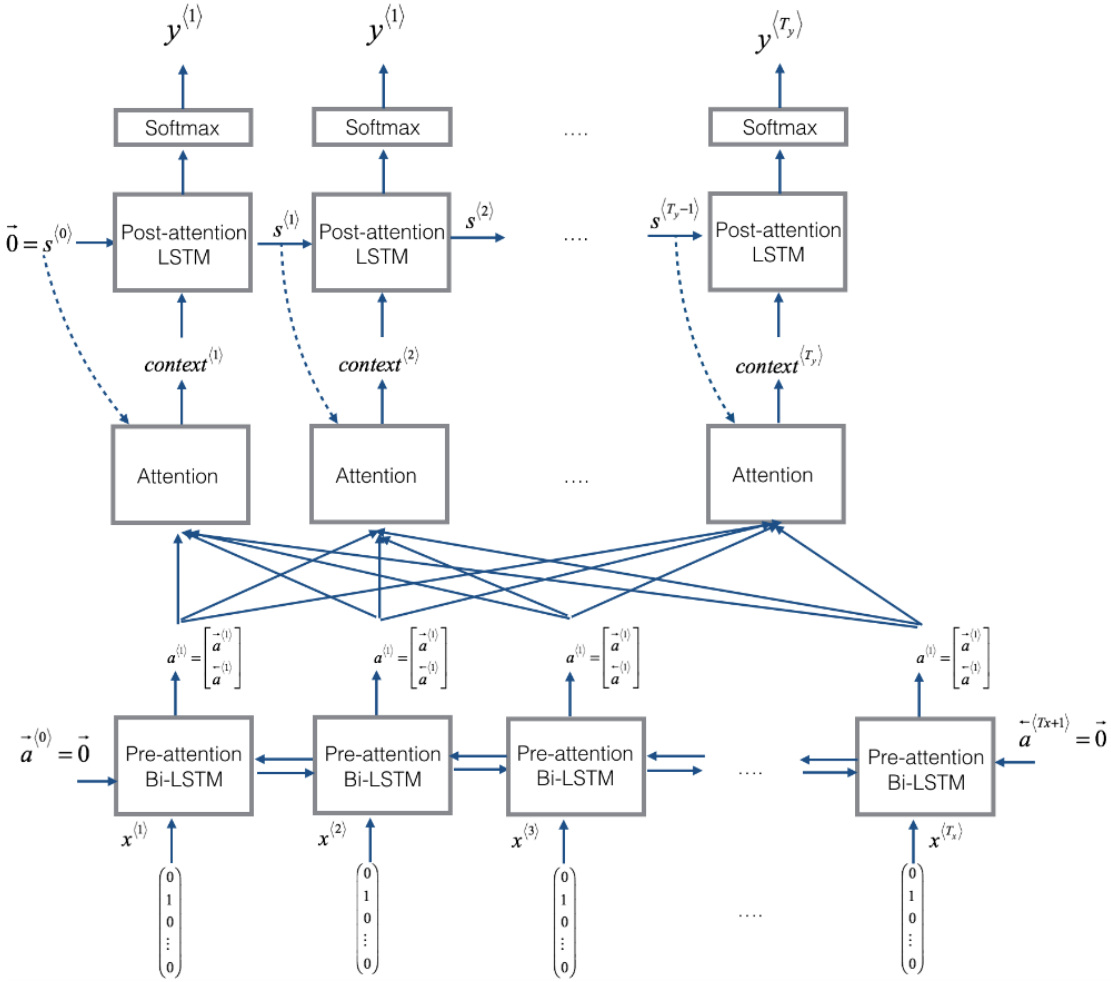



图 4-5 注意力机制在编码-解码器中

在先前的模型中，编码器部分的所有源输入信息都需要压缩到一个上下文向量中，这样使得上下文向量变的十分臃肿，而且如果源侧输入信息很长、很多的话，上下文向量的表达能力不一定能满足解码器解码所需的表达信息，为了避免这种臃肿表达，基于注意力的编码器-解码器结构能够避免这种臃肿。注意力层

首先会计算一个注意力向量，其长度等同于源侧输入的长度。注意力向量中的每个值都在 0-1 之间，并且整个向量中所有值的和是 1，之后对源侧输入的隐层状态计算一个加权和，得到加权源向量 w。在解码器解码的过程中，会在每个时间步计算一个新的注意力向量，从而在每个时间步得到一个新的加权源向量，使用此加权源向量作为解码器的输入。注意力层代码实现细节如图 4-6：

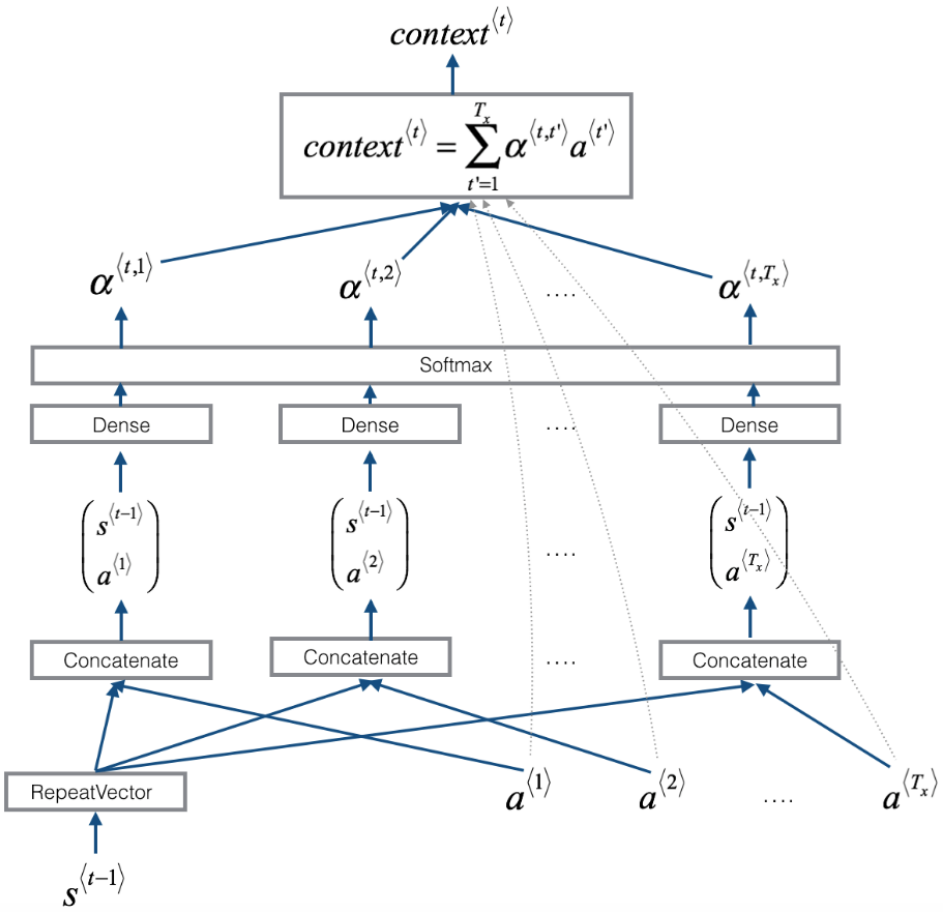



图 4-6 注意力层的实现

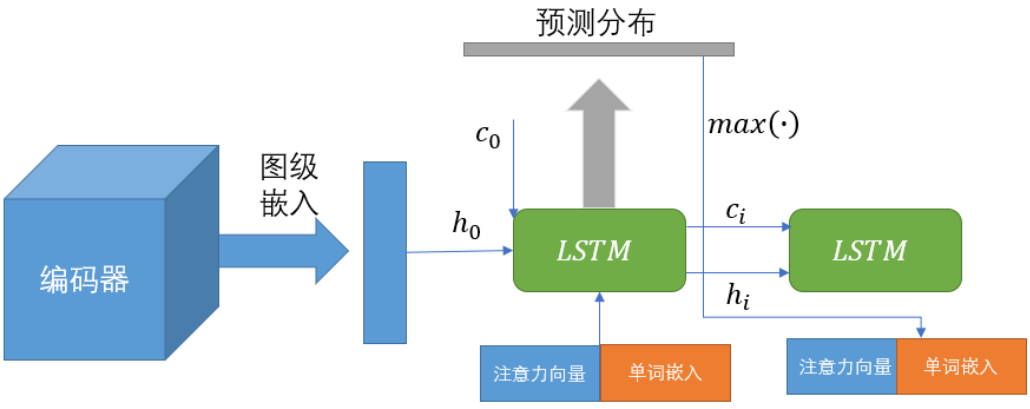



图 4-7 模型测试

在注意力层的计算中，需要编码器输出的所有隐层状态$H$和解码器在上一个时间步的隐层状态$s_{t-1}$，此层会输出一个元素在0~1间、元素和为1且长度等于源侧输入长度的注意力向量$a_t$。

直观地，此层会使用截至$t$时刻已经解码的信息$s_{t-1}$，和截至$t$时刻已经编码的所有信息$H$，去产生一个表示在$t$时刻我们需要去花费更多注意力去关注哪些源侧输入单词的注意力向量$a_t$，以便去预测解码的下一个单词$\hat{y}_{t+1}$。

首先通过编码器输出的所有隐层状态$H$和解码器在上一个时间步的隐层状态$s_{t-1}$计算一个能量值$energy$，因为编码器的隐层状态是$T_x$个张量，并且之前的解码器隐层状态也是单一的向量，首先需要做的就是将$t-1$时刻的解码器隐层状态复制$T_x$次，然后与编码器的隐层状态相连接得到连接向量，并将连接向量通过线性层$attn$，并使用$tanh$非线性激活，得到能量值$energy$，记为$E_t$：

$$E_t := tanh\big(attn(s_{t-1}, H)\big)$$

此公式可以看作每个编码器隐层状态和解码器隐层状态的匹配程度。

目前，如果线性层$attn$没有直接对连接向量进行降维的话，此时的$E_t$是个张量，即目前的维度是$[src_len，\ hidden_dim]$，为了将其维度降为$src_len$个单一值，需要一个维度是$[1，\ hidden_dim]$可学习的参数$v$，此时可获得$\hat{a}_t := vE_t$。

最后，为了确保注意力向量满足元素均在0~1间且和为1的约束，将$\hat{a}_t$通过$softmax$层：

$$a_t := softmax(\hat{a}_t)$$

此公式便让我们知道了该给不同的源侧输入单词赋予多少不同的注意力。

### 4.5.2 解码细节

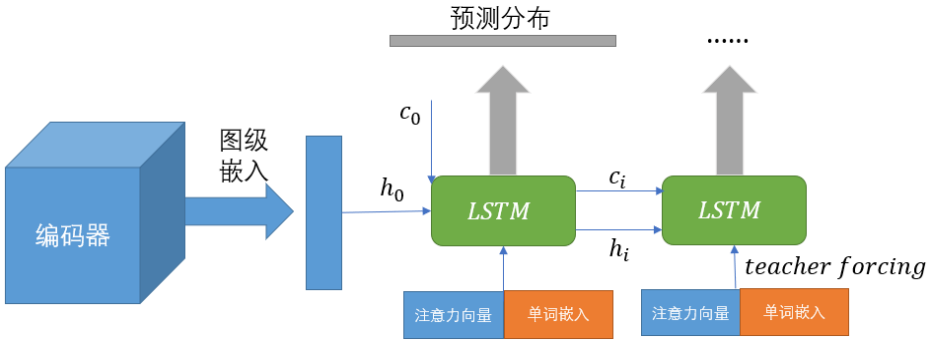



图 4-8 模型训练

初始长短时记忆网络时，隐层状态使用编码器的图级嵌入，细胞状态使用初始化0填充，当前时间步的输入使用第一个开始特殊符号<start>，当然实际上是该开始符号的嵌入表示，同时连接当前时刻的注意力向量；非初始化时，隐层状态使用解码器上一个时间步的输出隐层状态，细胞状态同理使用解码器上一个时间步的输出细胞状态，当前时刻的输入在训练时（如图 4-8）使用$teacher_forcing$，

即在每个时间步不适用模型预测输出，而是直接重新输入真值单词进行矫正，并连接当前时刻的注意力向量，在测试的时候（如图 4-7）使用的是上一时间步编码器的预测输出单词。搜寻预测过程使用贪婪搜索，其是在每个时刻的预测输出分布中寻找当前分布得分中的最大值作为结果。

### 4.6 评估模型

为了评估本工作模型的性能，选取了在机器翻译、图像描述中广泛使用了机器自动评估方法$BLEU$（双语互译质量评估辅助工具），其可以评估模型输出的文本与真值文本的相似性，根据流畅性和准确性来计算得分，其值在$0\sim1$之间，结果约接近$1$说明模型的性能越优。

$BLEU$总共有 4 个子单元，分别是$BLEU-1$、$BLEU-2$、$BLEU-3$和$BLEU-4$，其中$BLEU-1$通常用来表示目标句子和预测句子之间有多少单词对应，即不考虑流畅度，而只考虑准确性，$BLEU-2$、$BLEU-3$和$BLEU-4$在不同程度上体现了目标句子和预测输出句子在流畅度上的匹配相似程度，通常流畅度依次递增，即如果$BLEU-4$的得分很高的话，说明此模型的性能十分优越。

由于$BLEU-1$的计算公式等于

$$BLEU-1=\frac{候选句子有多少个单词出现在参考句子中}{候选句子的词汇数}$$

所以，$BLEU-1$的计算很容易受到常用词汇的干扰导致分数过高（如表 4-9 所示），因为$BLEU-1$的单词片段长度为$1$，易重叠。为了解决这种问题，提出了修正的计算方法$BLEU-2$、$BLEU-3$和$BLEU-4$。

| 候选句子 | *the* | *the* | *the* | *the* | *the* | *the* | *the* |
|---|---|---|---|---|---|---|---|
| 参考句子 | *the* | *cat* | *is* | *on* | *the* | *bed* | |

表 4-9

很明显，$BLEU-1$的计算误差出现在分子上，主要是因为候选句子中常用词居多，常用词也会频繁的出现在参考句子中，所以候选句子和参考句子的重叠度会增加，导致分子值会偏高。

为了解决这种问题，考虑将单词片段长度增加，如果单词片段长度不为$1$，而是增加至$2\sim4$，例如$BLEU-2$，将单词片段的组成部分由一个单词组成一个片段变成两个单词一个片段，这样就能放置$BLEU-1$时，常用词的干扰。

还有个问题，就是当候选句子长度较长的时候，$BLEU$的计算分母比较大，但是当候选句子长度比较短的时候，此时$BLEU$的计算分母比较小，这就致使计

算结果倾向于较大值,例如如表 4-10。

| 候选句子 | *the* | *cat* | | | | | |
|---|---|---|---|---|---|---|---|
| 参考句子 | *there* | *is* | *a* | cat | *on* | *the* | *bed* |

表 4-10

在这个例子中，计算结果为$\frac{1}{2}+\frac{1}{2}=1$，虽然结果为满分，但是显然这个候选句子并不是令人满意，甚至是一个不完整的句子。为了解决短候选句子容易取得高分的情况，必须添加短句惩罚。

$$\mathrm{BP}=\begin{cases}1 & ，如果c>r\\ e^{1-\frac{r}{c}} & ,如果c\le r\end{cases}$$

其中，$r$是参考翻译词数，$c$是候选翻译词数，$BP$代表短句惩罚值。

最后，将$BLEU-1$、$BLEU-2$、$BLEU-3$和$BLEU-4$组合起来，能更准确完善的反映模型的性能稳定性。四者得分基本上呈现下降的趋势（如图 5-10），也符合直观理解。在组合时，采用几何平均：

$$p_{ave}=exp\left(\frac{1}{4}*(\log p_1+\log p_2+\log p_3+\log p_4)\right)$$

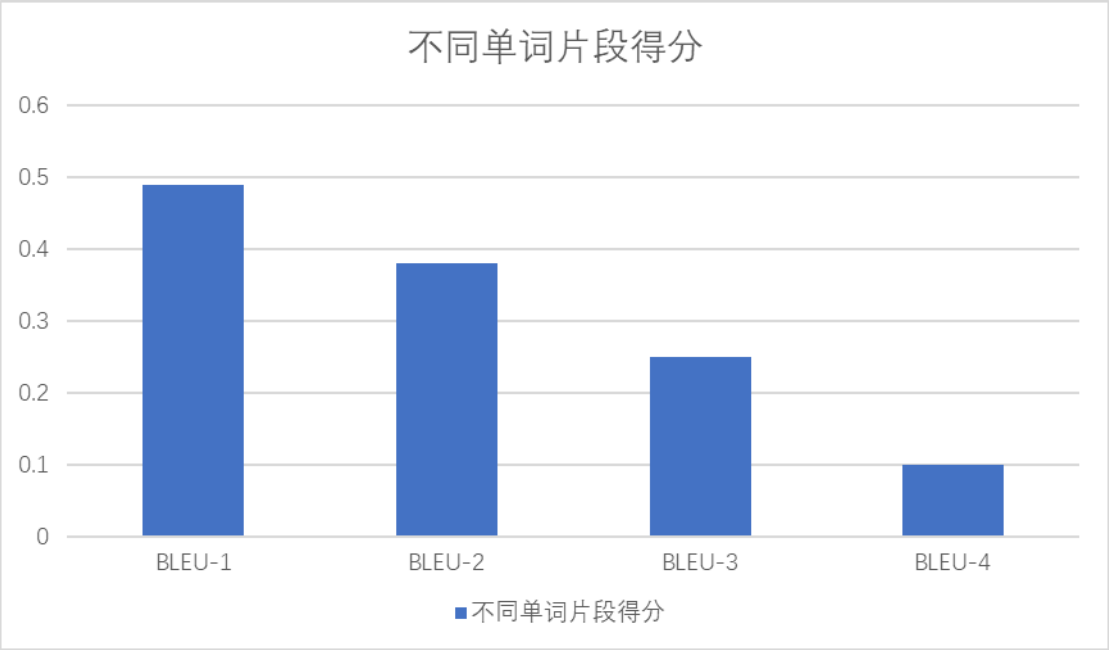



图 4-11

### 4.7 实验结果对比和演示

本实验编码器部分分别采用了序列模型、图模型，并在在不同情况下进行了实验结果比较，实验结果表明本工作的图到序列模型在性能上最优。

表 4-12 展示了聚合邻域信息的图模型比序列模型性能更好。Seq2Seq 模型代表序列模型，Seq2Seq+Attention 代表基于注意力的序列模型，G2Seq(NoEdge)代表无邻接矩阵的图到序列模型，G2Seq(NoEdge)+Attention 代表基于注意力的无邻接矩阵的图到序列模型，G2Seq 代表图到序列模型，G2Seq+Attention 代表基于注意力的图到序列模型；

| 模型 | BLEU |
|---|---|
| *Seq2Seq* | 48.53 |
| *Seq2Seq + Attention* | 50.69 |
| *G2Seq(NoEdge)* | 48.63 |
| *G2Seq(NoEdge) + Attention* | 52.44 |
| *G2Seq* | **49.56** |
| *G2Seq + Attention* | **52.45** |

表 4-12

| 模型 | BLEU |
|---|---|
| *Seq2Seq + Attention* | 50.69 |
| *GCN2Seq + Attention* | 51.03 |
| *GGNN2Seq + Attention* | 52.45 |
| *GAT2Seq + Attention* | **52.5** |
| *GGCNN2Seq + Attention* | **52.99** |

表 4-13

表 4-13 展示了不同的编码器所得到的性能也有所不同。Seq2Seq+Attention 代表编码器为序列模型，GCN2Seq+Attention 代表编码器为图卷积神经网络，GGNN2Seq+Attention 代表编码器为门控图神经网络，GAT2Seq+Attention 代表编

码器为图注意力网络，GGCNN2Seq+Attention 代表编码器为门图卷积神经网络。

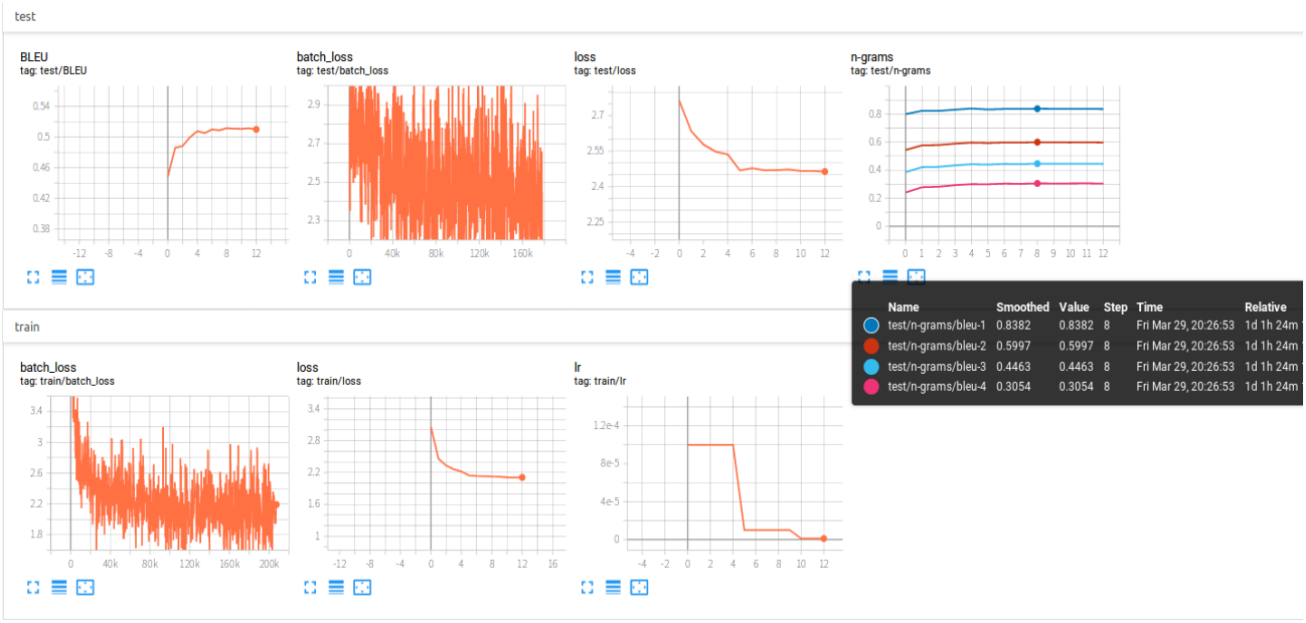



图 4-14 模型训练过程

图 4-14 展示了模型训练误差和测试误差。图一是模型评测得分，图二是测试批量误差下降曲线，图三是测试轮回下降曲线，图四是分离的模型评测得分，图五是训练批量下降曲线，图六是训练轮回下降曲线，图七是学习率下降曲线。

# 第五章 总结与展望

## 5.1 本文总结

本设计利用图神经网络扩展到序列解码任务。本项目引入了一种通用的端到端的图到序列神经编码器-解码器架构，该架构将输入图映射成节点及图嵌入，并使用基于注意力的长短时记忆力模型对这些矢量进行解码得到目标序列。本工作的方法首先使用改进的基于图的神经网络生成节点和图嵌入，该神经网络具有新颖的聚合策略，以在节点嵌入中结合近邻节点信息。且进一步引入了一种注意力机制，它将节点嵌入和解码序列对齐，以更好地处理大图并且使模型具有更好的解释性。

## 5.2 未来展望

将图神经网络编码器进行优化，可以针对稀疏矩阵进行计算优化，提升模型计算速度；针对解码器部分，可以使用转换器（Transformer）代替长短时记忆网络；解码搜索策略可以使用聚束搜索（Beam Search）代替贪婪搜索（Greedy Search）

# 参考文献


[1] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C].In Advances in neural information processing systems. 2012: 1097-1105.

[2] Mikolov T, Karafiát M, Burget L, et al. Recurrent neural network based language model[C].Eleventh annual conference of the international speech communication association. 2010.

[3] Krishna R, Zhu Y, Groth O, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations[J].In International Journal of Computer Vision, 2017, 123(1): 32-73.

[4] Xu K, Wu L, Wang Z, et al. Graph2seq: Graph to sequence learning with attention-based neural networks[J]. arXiv preprint arXiv:1804.00823, 2018.

[5] Beck D, Haffari G, Cohn T. Graph-to-sequence learning using gated graph neural networks[J]. arXiv preprint arXiv:1806.09835, 2018.

[6] Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate[J]. arXiv preprint arXiv:1409.0473, 2014.

[7] Xu K, Wu L, Wang Z, et al. SQL-to-Text Generation with Graph-to-Sequence Model[J]. arXiv preprint arXiv:1809.05255, 2018.

[8] Kipf T N, Welling M. Semi-supervised classification with graph convolutional networks[J]. arXiv preprint arXiv:1609.02907, 2016.

[9] Veličković P, Cucurull G, Casanova A, et al. Graph attention networks[J]. arXiv preprint arXiv:1710.10903, 2017.

[10] Li Y, Tarlow D, Brockschmidt M, et al. Gated graph sequence neural networks[J]. arXiv preprint arXiv:1511.05493, 2015.

[11] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[C].In Advances in neural information processing systems. 2017: 5998-6008.

[12] Scarselli F, Gori M, Tsoi A C, et al. The graph neural network model[J]. IEEE Transactions on Neural Networks, 2008, 20(1): 61-80.

[13] Wu Z, Pan S, Chen F, et al. A comprehensive survey on graph neural networks[J]. arXiv preprint arXiv:1901.00596, 2019.

[14] Litjens G, Kooi T, Bejnordi B E, et al. A survey on deep learning in medical image analysis[J]. Medical image analysis, 2017, 42: 60-88.

[15] Xu D, Zhu Y, Choy C B, et al. Scene graph generation by iterative message passing[C].In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 5410-5419.

[16] French M T, Robotham J S. Time inheritance scene graph for representation of media content: U.S. Patent 6,266,053[P]. 2001-7-24.

[17] Schlichtkrull M, Kipf T N, Bloem P, et al. Modeling relational data with graph convolutional networks[C].In European Semantic Web Conference. Springer, Cham, 2018: 593-607.

[18] Thomson S. Encoding and Decoding Graph Representations of Natural Language[D]. Carnegie Mellon University, 2019.
[19] Chung J, Gulcehre C, Cho K H, et al. Empirical evaluation of gated recurrent neural networks on sequence modeling[J]. arXiv preprint arXiv:1412.3555, 2014.
[20] Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization[J]. arXiv preprint arXiv:1409.2329, 2014.
[21] Sutskever I, Vinyals O, Le Q V. Sequence to sequence learning with neural networks[C].In Advances in neural information processing systems. 2014: 3104-3112.

# 致谢

光阴似箭，转眼间，大学四年的生活马上就要结束。在这四年中，我不仅学习了许多新的知识和专业技能、结交了五湖四海的新朋友，更重要的是通过四年的点点滴滴，学习了很多做人的道理和对生活的该有的幽默和积极向上的乐观心态！

在这里，我首先要感谢我的毕设指导老师王鹏。毕业设计是大学四年的终结考核，其综合考验一个人的耐力和解决应用问题的能力。王鹏老师为我选择了一个非常新颖、有挑战性的题目，这个题目激发了我的斗志，并且极强的锻炼了我独立解决问题的能力。当我遇到解决不了的问题或不知道如何下手进行下一步而迷茫时，王老师总能很好地指引我，为我指定一个明确的研究方向和思路，王老师为我提供了巨大的帮助，他丰富的专业知识和新奇的思路深深的折服了我。总之，毕业设计的完成离不开王老师的耐心指导，真诚的感谢他！

其次，我要感谢四年来所有教导过我、并且传授我知识和人生哲理的老师，你们不仅传道授业解惑，带领我了解计算机、进入计算机的世界，更在教授过程中传授一些你们所体会到的深刻的人生经验，从而让我能够更快的成熟，感谢各位老师！

然后，我要感谢我亲爱的同学们，尤其是班委们。在生活中，我们互相扶持；在学习中，我们互相帮助。我们一起认真上课、一起做实验、一起开心玩耍、一起运动健身、一起努力准备期末考试、一起为了各自以后的未来努力拼搏，总之，太多太多美好和值得回忆的事情，这四年，谢谢你们的陪伴！

最后，我要感谢我的家人，无论发生什么事情，你们总是我背后坚实的臂膀，为我答疑解惑、为我舒心，我爱你们，谢谢你们！同时感谢我的女朋友，谢谢你温情的陪伴，感谢帮我指出的论文排版问题。

感谢所有帮助我的人，在此祝老师们身体健康、万事如意，同学们工作顺利、生活开心、学业进步！

# 毕业设计小结

本科毕业设计是对大学四年自己成长过程中所学知识和综合能力的应用考验，是毕业前的一项重要工作和任务，它要求我们要认真、严谨并且坚持不懈的完成。经过四个月的努力，我终于完成了我的毕业设计，并且比开题报告时所要求的内容有所改进和提升。在刚开始拿到毕业设计的题目时，我甚至完全不懂如何去完成这项挑战性极强的问题，经过老师的一步步指导，我从懵懂到了解、再到有完整的认识，并且从代码实现上从零开始搭建框架，甚至刚开始入手时我都不熟悉所用的语言。不过还好，王老师的悉心有效指导给了我解决问题的希望，让我能坚持不懈、一步一个脚印的踏实走过来。经过这个有挑战性的项目的完整实现，极强的锻炼了我的工程代码能力和阅读英文文献的能力，并且对相关方向有了初步的认知。

在实现毕设的过程中，暴露了很多自身存在的问题，例如，英文阅读能力欠佳；在寻找和自己相关的方向的资料时效率欠佳；代码编码能力和框架搭建能力欠佳等。这个项目让我清楚的认识到自己的这些不足，让我能有机会、有针对性的去解决这些问题，同时提升自己在这些方面的能力。比如，我刚开始阅读英文文献的时候，好多专业词汇我不理解，通过更多的阅读和相关资料的对照，我能越来越熟练的阅读英文文献；我在对单词在嵌入表示的时候，没有利用误差反向传播来更新嵌入，导致模型效果一直很差，最后经过老师的指导才发现了这个问题，还有很多类似的大小问题，感谢毕业设计，让我能成为更好的的自己；感谢老师，能陪我一起成长。

这样一个自己从零开始参与负责的项目，检验了我大学四年所成长得到的知识与能力，并有效的提升了我解决问题的能力和个人的综合素养，让我对工程代码的实现有了较为清晰的认识，促使我在解决以后的工程问题时知道如何去查找自己所需的资料，并且如何去发现自己的实现过程中的问题。总是，这些让我明白要想不断提升自己，就得坚持不懈的去挑战自己，多做有难度的事情，不要局限于自己所学，多看、多读、多敲。在以后的研究生生活中，要更加努力的学习、更加拼搏。

总而言之，毕业设计让我学到很多东西，受益终生。