

第9组深度学习与应用期末报告

58120127 蔡英豪 58120301黄淇奥 58120305 吴悠

58120306 张雨琪 09020309李嘉楠 09020310程蓓蓓

摘要

随着深度学习在计算机视觉领域的广泛应用，基于深度学习方法的图像分类和图像描述研究取得了长足的进步。视频描述任务类似于图像描述。基于输入图像帧输出自然语言描述。不同的是，视频描述任务输入的是多个序列视频帧。

本文首先介绍了本次视频描述任务的背景和我们小组在视频描述任务中所涉及的模型，以及后续的具体实现。最后还补充了本小组在这学期课程中的扩展探究部分——扩散（diffusion）模型的简介以及其应用。

关键词：视频描述，深度学习，扩散模型

ABSTRACT

With the wide application of deep learning in the field of computer vision, research on image classification and image description based on deep learning methods has made great progress. The video description task is similar to image description. Outputs a natural language description based on an input image frame. The difference is that the input of the video description task is a sequence of video frames.

This article first introduces the background of this video description task and the models involved in our group's video description task, as well as the subsequent specific implementation. Finally, it supplements the extended research part of this semester's course by the group - an introduction to the diffusion model and its application.

KEY WORDS: video description, deep learning, Diffusion model

目 录

摘 要	I
ABSTRACT	II
目 录	III
第一章 引言	1
1.1 背景介绍	1
第二章 视频描述任务模型介绍	2
2.1 S2VT	2
2.1.1 模型简介	2
2.1.2 模型基本步骤	2
2.2 ResNet-152	3
2.2.1 ResNet 简介	3
2.3 VGGish	5
2.3.1 模型步骤	6
2.4 C3Dfeature	6
2.4.1 产生背景	6
2.4.2 模型网络结构	6
2.4.3 网络具体步骤	7
第三章 视频描述任务具体实现	8
3.1 特征嵌入	8
3.1.1 帧级特征	8
3.1.2 视频级别特征	8
3.1.3 音频特征	8
3.2 工具	8
3.2.1 ffmpeg	8
3.2.2 pydub + towhee 提取音频特征	9
3.3 特征融合	10
3.4 训练	12

3.5 总结	13
第四章 扩展部分	14
4.1 扩散模型	14
4.1.1 简介	14
4.1.2 扩散模型的应用	15
参考文献	19

第一章 引言

1.1 背景介绍

自 2012 年 AlexNet 提案 [1] 以来，深度学习在计算机视觉领域日益成为主流，并逐渐成为识别、语义分割、图像描述等大多数图像处理任务的首选方法。其中包括各种深度神经网络架构的变体，例如 ResNet[2]、CenterNet[3]。尽管针对不同的成像任务有许多不同的神经网络结构，但这些结构的基本框架都是相似的。或相同，其中大部分存在于卷积神经网络（CNN）、递归神经网络 [4]（recurrent neural networks, RNN）和注意力机制 [5]。

视频描述任务基于图像处理技术。在一个视频中，视频的长度在几秒到几十秒之间，包含几帧图像。该短视频使用自然语言描述来创建句子，该句子包含了该短视频的核心主题，视频描述旨在生成对视频图像的描述性描述。图像描述仅包含静态单帧图像，而视频描述包含动态多帧图像，具有前后场景和动作序列，类似于图像分类，将视频的语义内容分组为特定类别。

第二章 视频描述任务模型介绍

视频描述主要用到了 S2VT Model , ResNet-152, VGGish (语音特征), C3D feature 这几个模型。

2.1 S2VT

2.1.1 模型简介

S2VT 模型将通用的序列到序列模型应用到视频描述中，他是由两个 LSTM 网络叠加而成的，第一个 LSTM 是通过卷积神经网络提取到帧特征向量作为输入进行逐个编码，完成所有之后，就会逐个单词的生成一个句子（也就是先将视频拆分了图像，然后再对图像进行描述），并且在提高时序这件事上，计算了连续帧之间的光学流。这个模型避免了内容识别和句子生成的分离，直接学习输入视频与相应句子描述之间的映射关系，同时学习了一个以视觉特征为条件的语言模型。

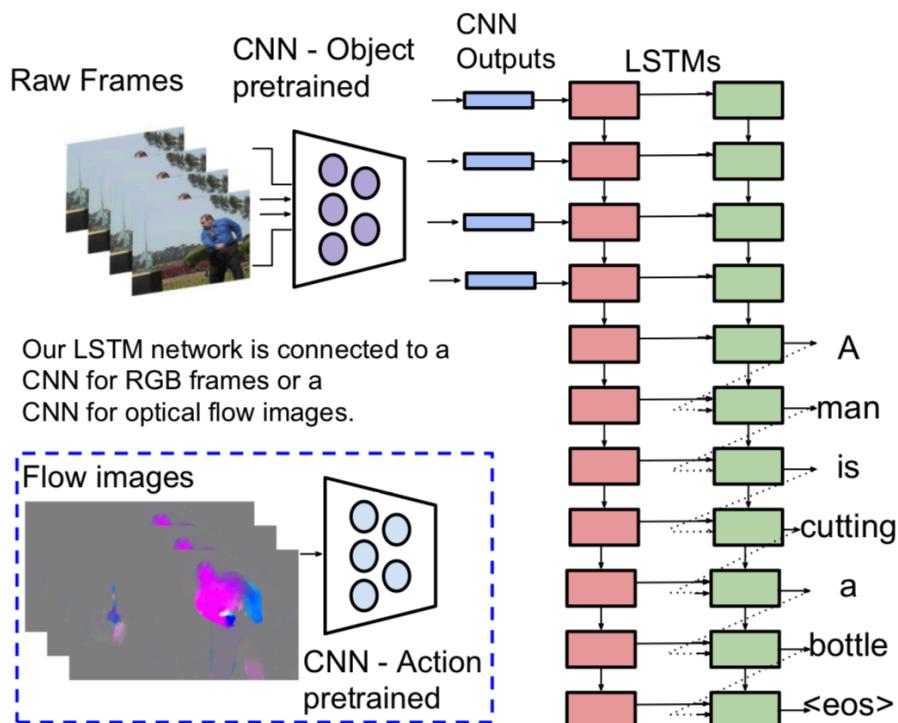


图 2-1 S2VT 网络结构

2.1.2 模型基本步骤

通过用 CNN 网络提取固定长度的图像特征，再通过 LSTM 网络将特征向量解码构成图像描述的单词序列。并且使用的是单一的 LSTM 在输入视频序列和输出文本序列之间

学习，做到了编码和解码参数共享。

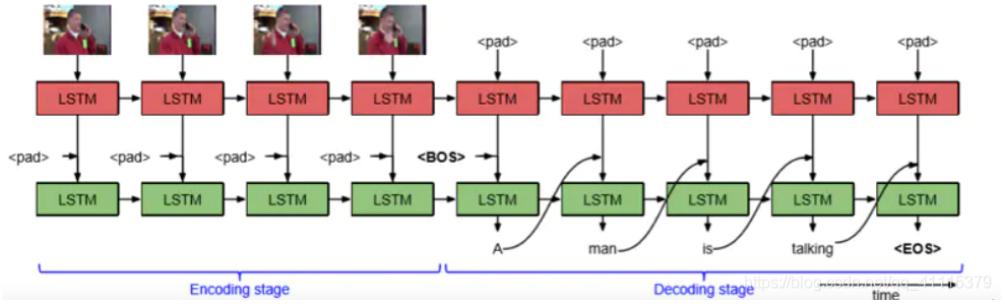


图 2-2 两层 LSTM 层

一开始先由顶层 LSTM 接受帧序列并进行编码，而第二层的 LSTM 接受第一层的隐含状态 h ，并将其与零填充符相连然后编码，并在所有帧都输出隐含状态后，第二层送入起始符，促使其开始将收到的隐藏状态解码成单词序列，解码阶段的话，就是在已经知道帧序列的隐藏状态及之前输出的单词条件下，求预测句子的对数似然性。

两层 LSTM 结构，顶层的对视觉特征进行建模，而第二层就是建立视觉序列隐藏状态表示的语言模型。

2.2 ResNet-152

ResNet-152 是图片分类任务中表现最好的神经网络之一，这次使用的是 resnet152 提取特征向量

2.2.1 ResNet 简介

ResNet (Residual Neural Network) 由微软研究院的 Kaiming He 等四名华人提出，通过使用 ResNet Unit 成功训练出了 152 层的神经网络，并在 ILSVRC2015 比赛中取得冠军，在 top5 上的错误率为 3.57%，同时参数量比 VGGNet 低，效果非常突出。ResNet 的结构可以极快的加速神经网络的训练，模型的准确率也有比较大的提升。同时 ResNet 的推广性非常好，甚至可以直接用到 InceptionNet 网络中。

ResNet 的主要思想是在网络中增加了直连通道，Highway Network 的思想。此前的网络结构是将输入做一个非线性变换，而 Highway Network 则允许保留之前网络层的一定比例的输出。ResNet 的思想和 Highway Network 的思想也非常类似，允许原始输入信息直接传到后面的层中，如下图所示：

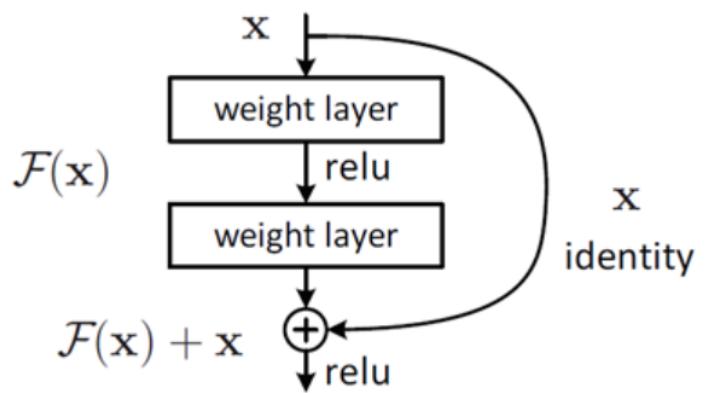


图 2-3 残差网络的创新

ResNet 网络结构

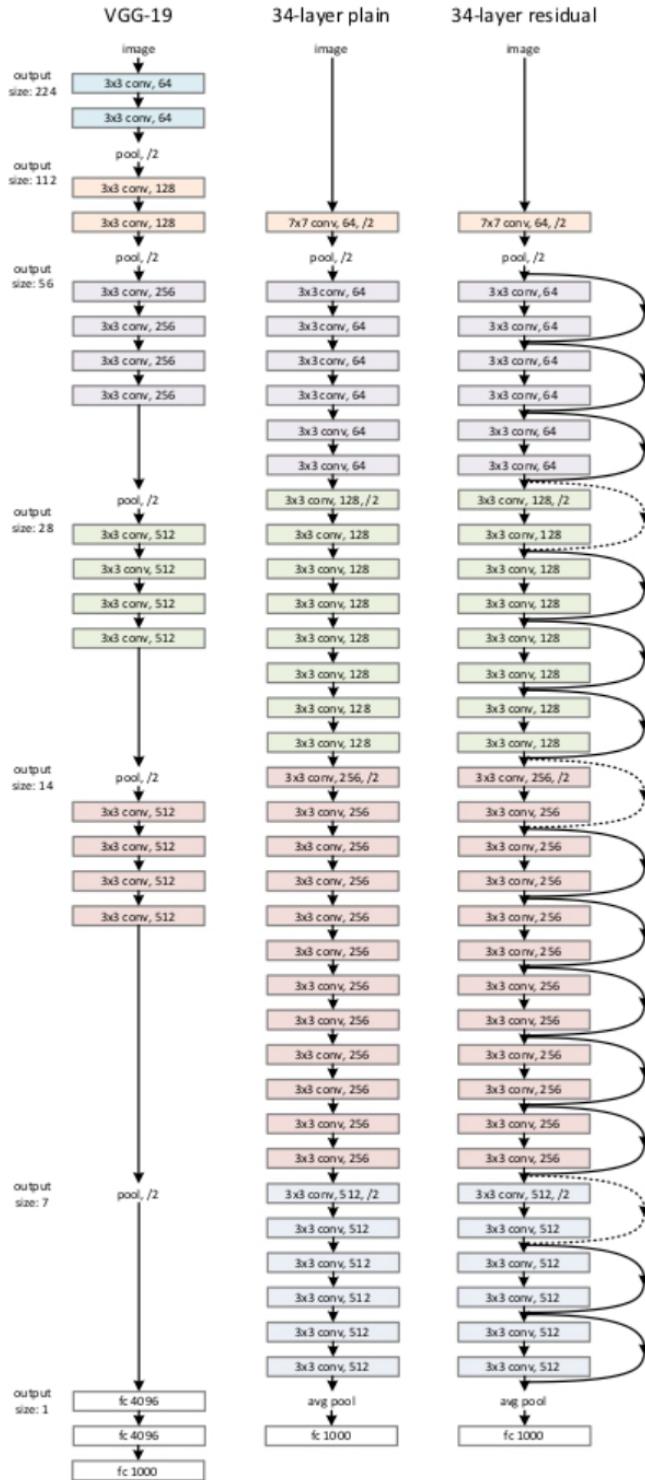


图 2-4 残差网络结构与其他网络结构对比

2.3 VGGish

谷歌在 2017 年公开了大规模音频数据集 AudioSet，包含了大约 210 万个长度为 10 秒的声音片段和 527 个标签。随即谷歌使用该数据集进行预训练，最终得到 VGGish 模型用

于音频的特征提取。该模型作为特征提取器，将音频输入特征转化为具有语义和有意义的 128 维 high-level 的特征向量，而 128 维 high-level 特征向量可以作为下游模型的输入。

2.3.1 模型步骤

1. 将音频重采样为 16kHz 单声道音频；
2. 使用 25 ms 的 Hann 时窗，10 ms 的帧移对音频进行短时傅里叶变换得到频谱图；
3. 通过将频谱图映射到 64 阶 mel 滤波器组中计算 mel 声谱；
4. 计算 $\log(\text{mel-spectrum} + 0.01)$ ，得到稳定的 mel 声谱，所加的 0.01 的偏置是为了避免对 0 取对数；
5. 然后这些特征被以 0.96s 的时长被组帧，并且没有帧的重叠，每一帧都包含 64 个 mel 频带，时长 10ms（即总共 96 帧）。

VGGish 模型输出数据格式为 [nums_frames,128]，其中 nums_frames 为帧长，nums_frames= 音频时长/0.96。

2.4 C3Dfeature

2.4.1 产生背景

卷积神经网络（CNN）被广泛应用于计算机视觉中，包括分类、检测、分割等任务。这些任务一般都是针对图像进行的，使用的是二维卷积（即卷积核的维度为二维）。而对于基于视频分析的问题，2D convolution 不能很好得捕获时序上的信息，因此 3D 卷积就被提出来了。

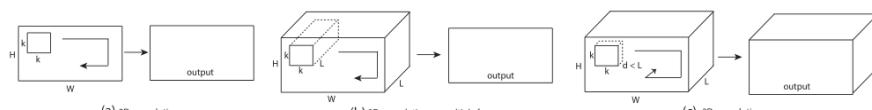


Figure 1. **2D and 3D convolution operations.** a) Applying 2D convolution on an image results in an image. b) Applying 2D convolution on a video volume (multiple frames as multiple channels) also results in an image. c) Applying 3D convolution on a video volume results in another volume, preserving temporal information of the input signal.
<https://blog.csdn.net/hehuaiyuyu>

图 2-5 2D 和 3D 网络的运作

2.4.2 模型网络结构



Figure 3. **C3D architecture.** C3D net has 8 convolution, 5 max-pooling, and 2 fully connected layers, followed by a softmax output layer. All 3D convolution kernels are $3 \times 3 \times 3$ with stride 1 in both spatial and temporal dimensions. Number of filters are denoted in each box. The 3D pooling layers are denoted from pool1 to pool5. All pooling kernels are $2 \times 2 \times 2$, except for pool1 is $1 \times 2 \times 2$. Each fully connected layer has 4096 output units.
<https://blog.csdn.net/hehuaiyuyu>

图 2-6 C3D 网络结构

8 个卷积层、5 个池化层、两个全连接层，以及一个 softmax 输出层；

所有 3D 卷积核均为 $3 \times 3 \times 3$ ($d \times k \times k$, d 为时间深度)，步长为 $1 \times 1 \times 1$ ；

为了在早期阶段保留更多的时间信息，设置 pool1 核大小为 $1 \times 2 \times 2$ 、步长 $1 \times 2 \times 2$ （时间深度为 1 时，会单独在每帧上进行池化，大于 1 时，会在时间轴上，也就是多帧之间进行池化）其余所有 3D 池化层均为 $2 \times 2 \times 2$ ，步长为 $2 \times 2 \times 2$ ；

每个全连接层有 4096 个输出单元。

2.4.3 网络具体步骤

从每个训练视频中随机取出 2 秒长的 5 个片段，调整帧大小为 128×171 （大约为 UCF-101 一半分辨率）；

随机裁剪成 $16 \times 112 \times 112$ 的片段（shape: $[N, C, nframe, H, W] \rightarrow [N, 3, 16, 112, 112]$ ，16 帧片段非重叠），形成抖动，以 50% 的概率随机翻转；

使用 SGD 优化器，batch size 为 30，初始学习率为 0.003，每 150K 次迭代除以 2，优化在 1.9M 迭代（约 13 epochs）停止

第三章 视频描述任务具体实现

3.1 特征嵌入

3.1.1 帧级特征

Resnet-152 也可以用 inception 模块获取

3.1.2 视频级别特征

C3D feature

3.1.3 音频特征

VGGish

3.2 工具

3.2.1 ffmpeg

ffmpeg 是由法布里斯·贝拉（这个人是一个传奇程序员，除了 ffmpeg 还提出了计算圆周率最快的算法，贝拉公式）主导开发的。FFmpeg 全称是 Fast Forward Mpeg，是一套可以用来记录、转换数字音频、视频，并能将其转化为流的开源计算机程序。

命令行工具，我们在 python 中使用的时候使用 subprocess 模块来调用

```
def extract_frames(video, dst):
    with open(os.devnull, "w") as ffmpeg_log:
        if os.path.exists(dst):
            print(" cleanup: " + dst + "/")
            shutil.rmtree(dst)
        os.makedirs(dst)
        video_to_frames_command = ["ffmpeg",
                                    # (optional) overwrite output file if it exists
                                    '-y',
                                    '-i', video, # input file
                                    '-vf', "scale=400:300", # input file
                                    '-qscale:v', "2", # quality for JPEG
                                    '{0}/%06d.jpg'.format(dst)]
        subprocess.call(video_to_frames_command,
                      stdout=ffmpeg_log,
                      stderr=ffmpeg_log)
```

图 3-1 命令行截图

ffmpeg 的作用就是对视频进行采样，根据设定好的采样率从视频中抽帧（我们设定的是每个视频抽 40 帧）。

采样结束之后使用预训练模型进行特征提取，得到每个视频的帧级特征：

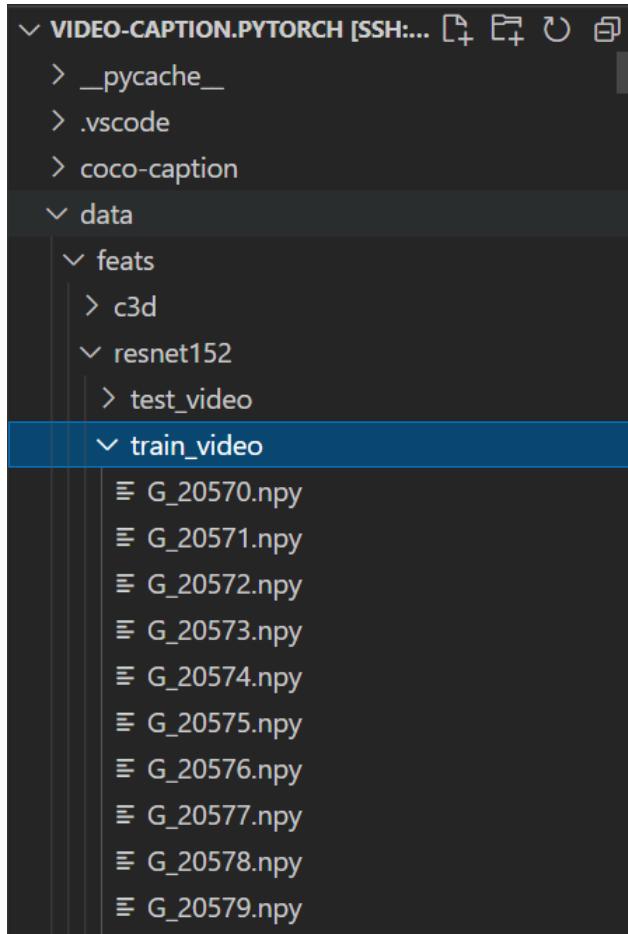


图 3-2 视频特征提取

3.2.2 pydub + towhee 提取音频特征

pydub

pydub 是一个开源的音频处理库，提供了简洁的高层接口，极大的扩展了 python 处理音频文件的能力，可以满足大多数情况下的音频处理需求。

由于 VGGish 预训练模型只能接受 wav 文件，而数据集里面的音频是 aac 文件所以需要转码。转码这个功能再各 APP 或者在线网站大多都是要收费的，而且很少有批量转换的功能。但是使用 pydub 只需要三行代码就可以搞定：

```
1 from pydub import AudioSegment
2 song = AudioSegment.from_mp3("G_20570.aac")
3 song.export("G_20570.wav", format="wav")
[2] ✓ 0.3s
... <_io.BufferedRandom name='G_20570.wav'>
```

图 3-3 音频处理

Towhee

Towhee 是一个开源的 embedding 框架，包含丰富的数据处理算法与神经网络模型。通过 Towhee，能够轻松地处理非结构化数据（如图片、视频、音频、长文本、分子结构等），完成原始数据到向量的转换。

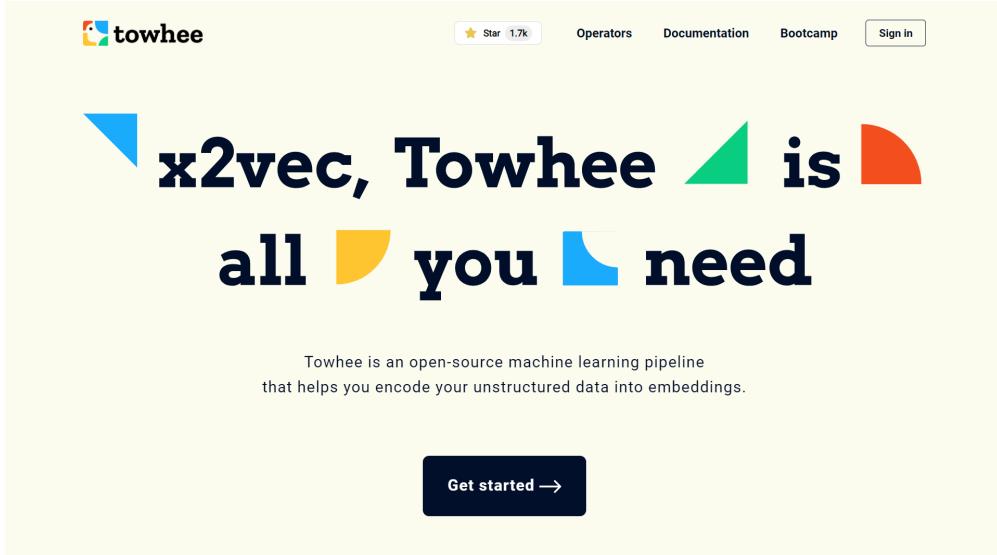


图 3-4 Towhee

这个首页的标语，“ToWhee is all you need !”)

Towhee 提供跨 5 个领域 (CV、NLP、多模态、音频、医学)、15 种任务、140 个模型架构、700 个预训练的 SOTA 模型 (例如：BERT、CLIP、ViT、SwinTransformer、MAE、data2vec 等)。

它使用起来极其简单，只需要几行代码就能实现特征的提取。

```
1 from towhee import pipeline
2 p = pipeline("towhee/audio-embedding-vggish")
3 vec = p("G_20570.wav")
```

Python

H:\anaconda\envs\start\lib\site-packages\tqdm\auto.py:22: TqdmWarning: IProgress not found.
Please update jupyter and ipywidgets. See
https://ipywidgets.readthedocs.io/en/stable/user_install.html

```
from .autonotebook import tqdm as notebook_tqdm
```

图 3-5 音频特征提取

3.3 特征融合

多模态特征融合的方法大体分为三种：前端融合、中间融合和后端融合。其中：

前端融合指的是将多个独立的数据集融合成一个单一的特征向量，然后输入到机器学

习分类器中。多模态前端融合方法常常与特征提取方法相结合以剔除冗余信息，如主成分分析（PCA）、最大相关最小冗余算法（mRMR）、自动解码器（Autoencoders）等。本人研究的是使用深层联合自编码模型，将三种模态的特征使用三层线性层将维度转化为同一维度，然后相加，最后将三者进行还原回去。

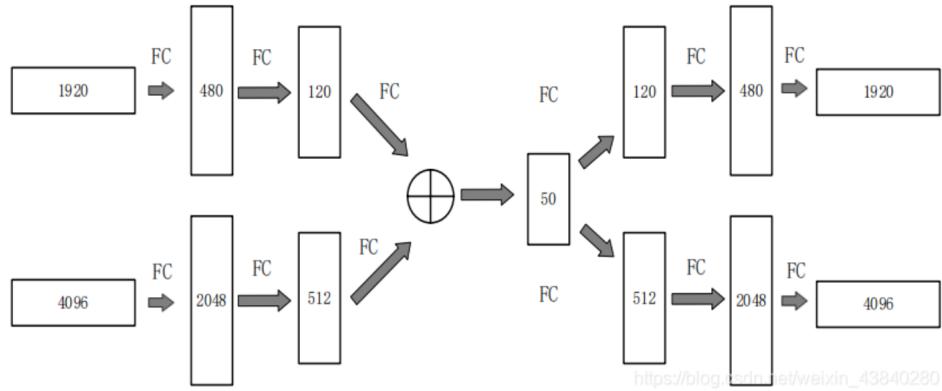


图 3-6 特征融合过程

中间融合指的是将不同的模态数据先转化为高维特征表达，再于模型的中间层进行融合。以神经网络为例，中间融合首先利用神经网络将原始数据转化成高维特征表达，然后获取不同模态数据在高维空间上的共性。在问答对话中有 MFB 方法（github 地址），它针对文本和图像两种模态，先将每个模态特征转化为相同维度的高维向量，然后进行逐元素相乘，最后进行 sum pooling 操作。

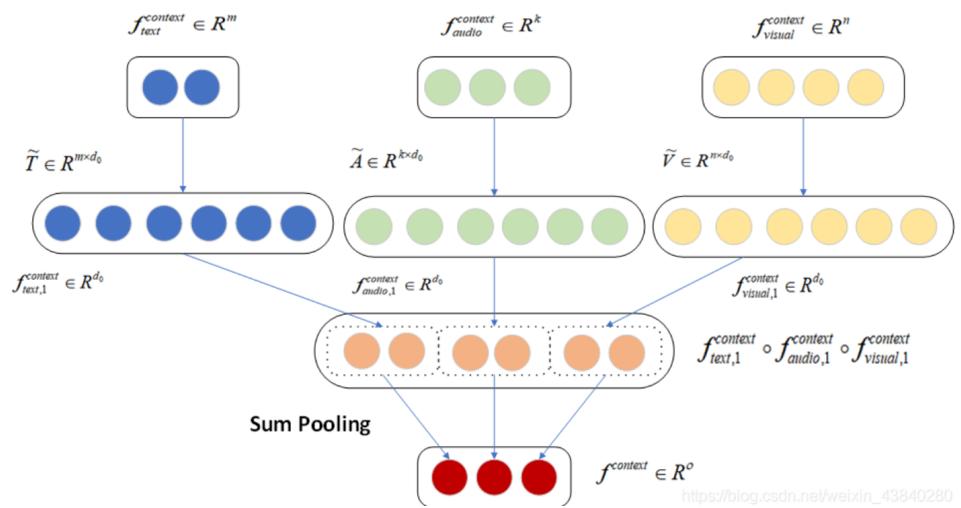


图 3-7 中间融合

后端融合指的是将不同模态数据分别训练好的分类器输出打分（决策）进行融合。常见

的后端融合方式包括最大值融合 (max-fusion)、平均值融合 (averaged-fusion)、贝叶斯规则融合 (Bayes' rule based) 以及集成学习 (ensemble learning) 等。

我们采用的是**中间融合**

3.4 训练

获取特征之后我们用 S2VModel 建模，进行了 3000 epoch 的训练

loss 曲线如下：

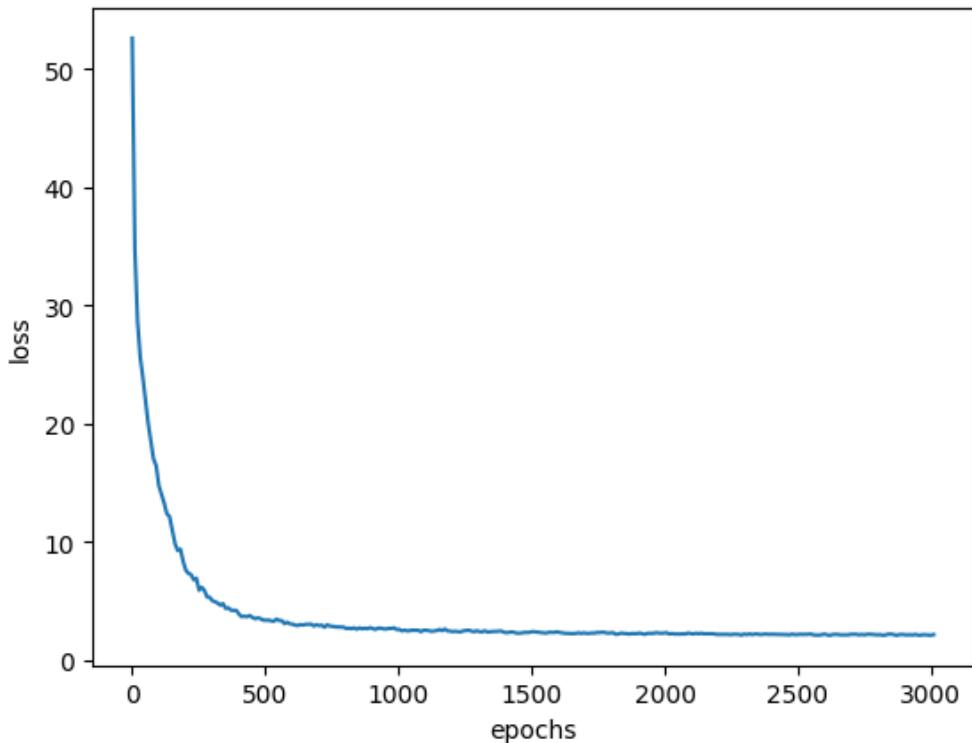


图 3-8 loss 函数

可以看到最终是一个收敛的状态。

然而，用最终模型去获取的结果非常不理想：

2	0.67693	answer.zip	11/26/2022 03:33:34	118099	Finished	
---	---------	------------	---------------------	--------	----------	--

图 3-9 初步结果

在和其他组交流之后，我们发现同样是 S2VTModel，其他组的结果要比我们好很多，模型的上限远不止这个结果。

经过分析我们认为这是因为训练代数过多导致的模型过拟合。于是我们在训练数据中划分了一部分数据作为验证集 (200 samples)，每隔 10 epochs 检查一下在验证集上的指标，

如果开始下降则停止训练。

最终我们的训练在 100epochs 停止，我们使用 80epochs 得到的模型取得了较好的效果。

9	0.87619	answer_80.zip	12/06/2022 12:18:20	230945	Finished	✓	+
---	---------	---------------	---------------------	--------	----------	---	---

图 3-10 最终结果

3.5 总结

用一个简单传统的模型，通过多种特征的融合和各种 trick 取得了还不错的结果

第四章 扩展部分

4.1 扩散模型

4.1.1 简介

扩散模型 (Diffusion Models) 定义了扩散步骤的马尔可夫链，通过添加随机噪声将输入的原始数据变成纯高斯噪声，然后学习逆向扩散过程以从噪声中构造所需的数据样本，分为前向过程和逆向（推断）过程两部分。与 VAE 或流模型不同，扩散模型是通过固定过程学习的，并且潜在变量的维度与原始数据相同。扩散模型是一类基于似然度的模型，最近被证明可以生成高质量的图像，同时提供了理想的属性，如更高的分布覆盖率，稳定的训练目标和更好的可扩展性。这些模型通过逐步去除信号中的噪声来生成样本，其训练目标可以表示为一个重新加权的变分下界。

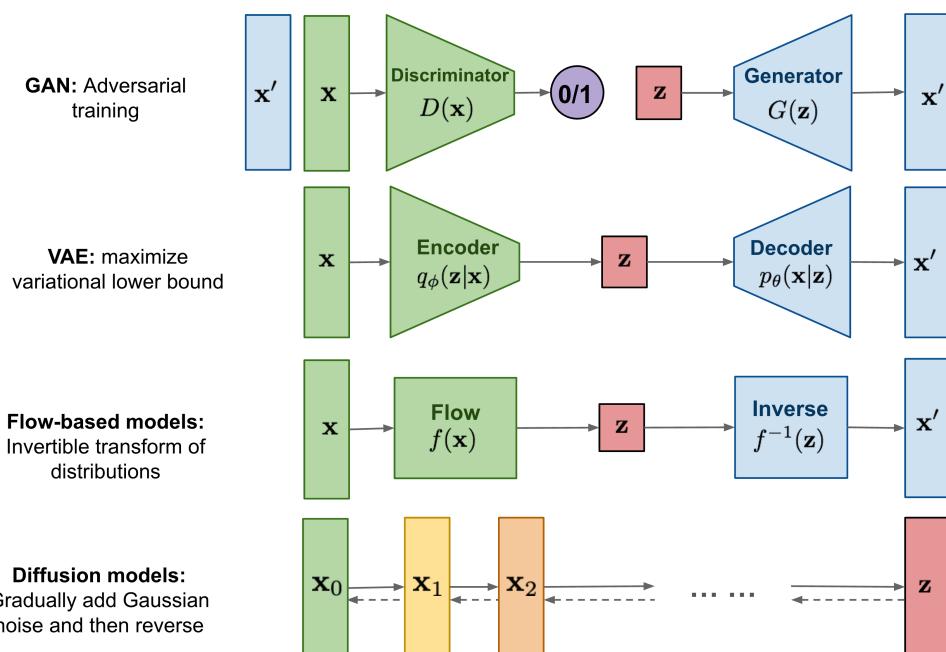


图 4-1 Diffusion Model 与其他模型的对比

扩散模型主要有两个过程：扩散 (diffusion) 和采样 (sampling)

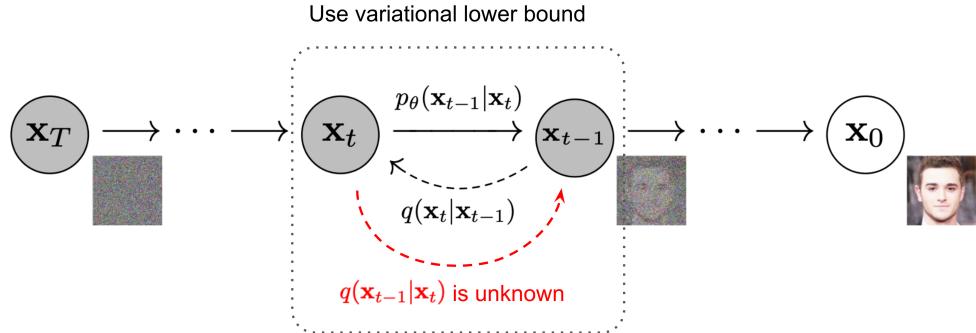


图 4-2 Diffusion Model 与其他模型的对比

4.1.2 扩散模型的应用

自然语言处理

在可控自然语言生成任务上，利用连续扩散模型，对预训练的语言生成模型进行可插拔的操控。

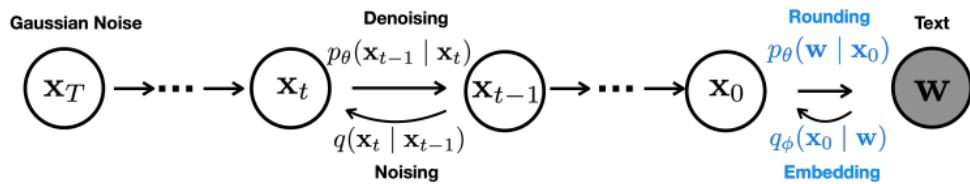


图 4-3 Diffusion Model 与其他模型的对比

	Semantic Content ctrl ↑	Content lm ↓	Parts-of-speech ctrl ↑	Parts-of-speech lm ↓	Syntax Tree ctrl ↑	Syntax Tree lm ↓	Syntax Spans ctrl ↑	Syntax Spans lm ↓	Length ctrl ↑	Length lm ↓
PPLM	9.9	5.32	-	-	-	-	-	-	-	-
FUDGE	69.9	2.83	27.0	7.96	17.9	3.39	54.2	4.03	46.9	3.11
Diffusion-LM	81.2	2.55	90.0	5.16	86.0	3.71	93.8	2.53	99.9	2.16
FT-sample	72.5	2.87	89.5	4.72	64.8	5.72	26.3	2.88	98.1	3.84
FT-search	89.9	1.78	93.0	3.31	76.4	3.24	54.4	2.19	100.0	1.83

Table 2: Diffusion-LM achieves high success rate (ctrl ↑) and good fluency (lm ↓) across all 5 control tasks, outperforming the PPLM and FUDGE baselines. Our method even outperforms the fine-tuning oracle (FT) on controlling syntactic parse trees and spans.

图 4-4 实验结果 1

	Semantic Content + Syntax Tree			Semantic Content + Parts-of-speech		
	semantic ctrl ↑	syntax ctrl ↑	lm ↓	semantic ctrl ↑	POS ctrl ↑	lm ↓
FUDGE	61.7	15.4	3.52	64.5	24.1	3.52
Diffusion-LM	69.8	74.8	5.92	63.7	69.1	3.46
FT-PoE	61.7	29.2	2.77	29.4	10.5	2.97

Table 4: In this experiment, we compose semantic control and syntactic control: Diffusion-LM achieves higher success rate (ctrl ↑) at some cost of fluency (lm ↓). Our method outperforms both FUDGE and FT-PoE (product of experts of two fine-tuned models) on control success rate, especially for the structured syntactic controls (i.e. syntactic parse tree and POS).

图 4-5 实验结果 2

作者们在情感控制，可控语法生成等任务上开展了实验，和 PPLM, FUDGE 等可插拔式方法进行对比，可以发现 Diffusion-LM 相比之前的同类方法有极为显著的提升，特别是在部分任务上，甚至可以达到和微调相似的结果。

蛋白质结构生成

尽管蛋白质结构预测已经取得了非常好的成绩，但要从神经网络中直接生成多结构多样又新颖的蛋白质结构仍然很困难。作者想到用基于扩散的生成模型来挑战这一任务，并通过镜像蛋白质自然折叠过程来设计蛋白质主链结构。具体来说，就是将蛋白质主链结构看成一系列连续的角度，这些角度会捕捉组成氨基酸残基的相对方向。进而通过从随机、未折叠状态到稳定折叠结构的去噪就可以生成新结构。这一设计不仅可以反映蛋白质如何在生物学上扭曲成能量上有利的结构，这种表示的固有位移和旋转不变性也可以极大地减轻模型对复杂等变网络的需要。

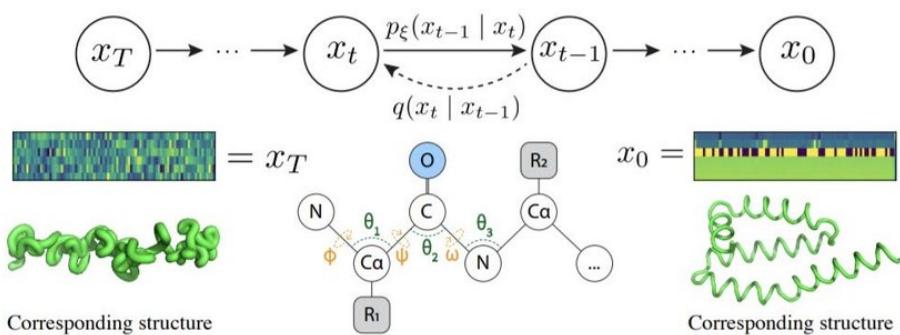


Figure 1: We perform diffusion on six angles as illustrated in the schematic in the bottom center (also defined in Table 1). Three of these are dihedral torsion angles (orange), and three are bond angles (green). We start with an experimentally observed backbone described by angles x_0 and iteratively add Gaussian noise via the forward noising process q until the angles are indistinguishable from the wrapped Gaussian at x_T . We use these examples to learn the “reverse” denoising process p_ξ .

图 4-6 蛋白质结构模型预测

在实现上，作者仅用一个简单的 transformer 作为 backbone 就训练出了一个去噪扩散概率模型。最终证明它可以无条件地生成高度真实的蛋白质结构，其复杂性和结构模式类似于天然蛋白质的结构模式。如下拉氏图（一种专门用于检测蛋白质构象是否合理的图）所示，左右分别为测试集和生成的蛋白质主干的 (ϕ, ψ) 二面角，三个主要结构元素、以及一些不太常见的角度组合都在他们用扩散模型生成的主干中得到了呈现。

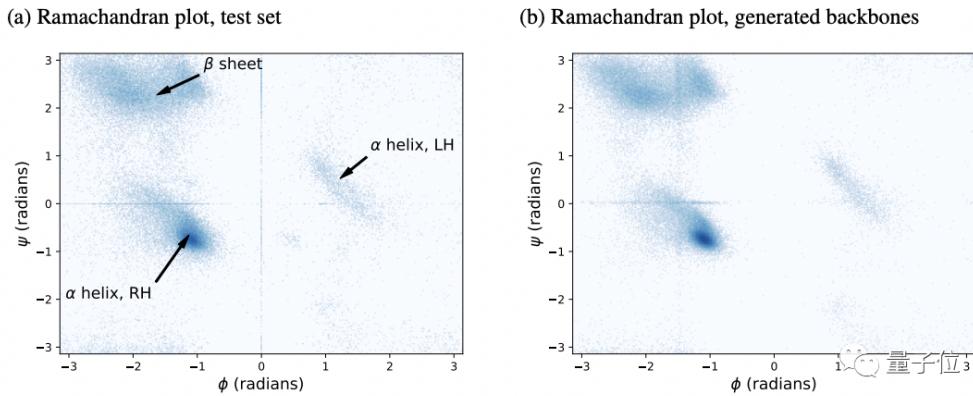


图 4-7 蛋白质的拉氏图

计算机视觉

DALL-E 2 OpenAI 的 Guided Diffusion 提出了一种简单有效的类别引导的扩散模型生成方式：在逆向过程的每一步，用一个分类网络对生成的图片进行分类，再基于分类分数和目标类别之间的交叉熵损失计算梯度，用梯度引导下一步的生成采样。这个方法一个很大的优点是，不需要重新训练扩散模型，只需要在前馈时加入引导既能实现相应的生成效果。在 Guided Diffusion 中，每一步逆向过程里通过引入朝向目标类别的梯度信息，来实现针对性的生成。这个过程其实和基于优化的图像生成算法（即固定网络，直接对图片本身进行优化）有很大的相似之处。这就意味着之前很多基于优化的图像生成算法都可以迁移到扩散模型上。换一句话说，我们可以轻易地通过修改 Guided Diffusion 中的条件类型，来实现更加丰富、有趣的扩散生成效果。在 Semantic Guidance Diffusion (SGD) 中，作者就将类别引导改成了基于参考图引导以及基于文本引导两种形式，通过设计对应的梯度项，实现对应的引导效果，实现了不错的效果。

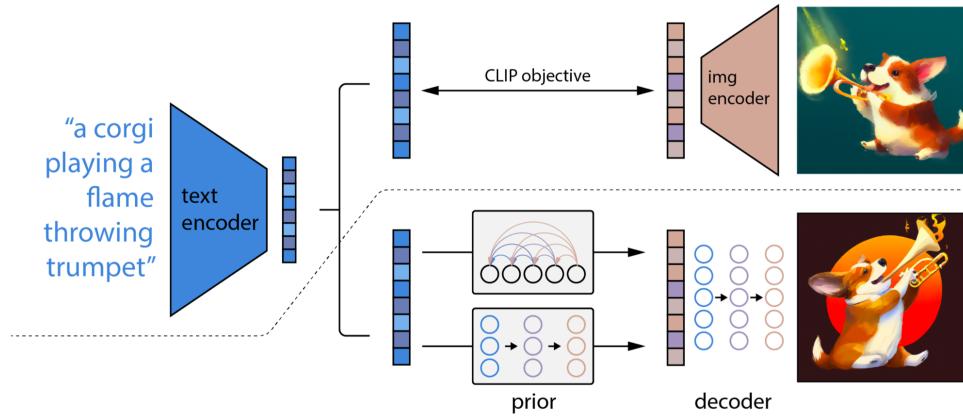


图 4-8 图像预测过程

除了分类引导，还可以有文本，图像等多种引导方式。基于文本条件的图像生成，即希望生成的图像符合文本的描述。在逆向过程中，每个迭代步要对有噪声的图像和文本计算 embedding 相似度，作为引导。

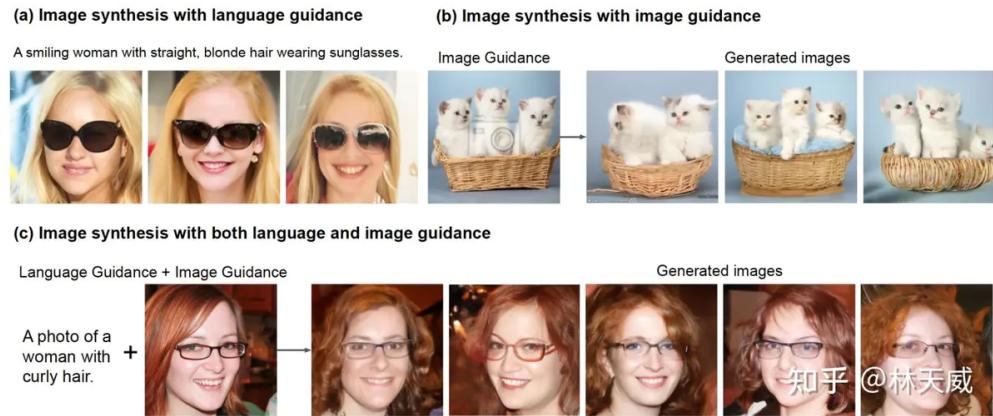


图 4-9 图像预测

参考文献

- [1] CHIDAMBER S R, KEMERER C F. A metrics suite for object oriented design[J]. Software Engineering IEEE Transactions on, 1994, 20(6): 476 - 493.
- [2] SUBRAMANYAM R, KRISHNAN M S. Empirical analysis of ck metrics for object-oriented design complexity: Implications for software defects[J]. IEEE Transactions on Software Engineering, 2003, 29(4): 297-310.
- [3] KRIZHEVSKY A, SUTSKEVER I, HINTON G. Imagenet classification with deep convolutional neural networks[J]. Advances in neural information processing systems, 2012, 25(2).
- [4] MIKOLOV T, KARAFIÁT M, BURGET L, et al. Recurrent neural network based language model[C]// Interspeech, Conference of the International Speech Communication Association, Makuhari, Chiba, Japan, September. 2015.
- [5] HE K, ZHANG X, REN S, et al. Deep residual learning for image recognition[J]. IEEE, 2016.
- [6] VASWANI A, SHAZER N, PARMAR N, et al. Attention is all you need[J]. CoRR, 2017, abs/1706.03762.
- [7] HERSEY S, CHAUDHURI S, ELLIS D P W, et al. CNN architectures for large-scale audio classification [J/OL]. CoRR, 2016, abs/1609.09430. <http://arxiv.org/abs/1609.09430>.