Computer Vision Authors：丁佑龙 孙祺华

MaskGIT Milestone Report

1 About the Problem

1.1 Introduction

在这个实验中，我们将要着眼于 MaskGIT(Masked generative image transformer) 来完成图像生成的任务。MaskGIT 是一种使用双向 transformer 解码器的新型图像合成模型，在训练期间，其通过关注各个方向的 token 来学习预测随机掩码 token。在推理阶段，模型首先同时生成图像的所有 token，然后以上一次生成为条件迭代地细化图像。实验表明，MaskGIT 在 ImageNet 数据集上显著优于 SOTA transformer 模型，并将自回归解码的速度提高了 64 倍。

本实验意在从某一视角复现 MaskGIT 的论文实验结果，总体规划为模型搭建，模型的代码实现，数据集的处理和训练，以及最后的模型评估。

1.2 Problem Statement

我们处理的问题是将残缺不全的照片传入一个训练好的神经网络（所谓残缺不全，指在图像中的一些矩形区域缺失内容，呈现为一片空白），神经网络会基于对残缺不全的照片的理解，生成一个语义相近的照片，或许会与原照片在未被遮掩的地方完全一致，或许不会，一切要基于之后的研究推进，但肯定的是，会满足语义相近，猫会是猫，狗会是狗。我们预期使用 tiny-imagenet-200 数据集，具体的基于数据集的操作系统还需要在之后定下具体计划后构建。我们将使用 Frechet Inception Distance 对生成图像的质量进行评估。

2 Technical Approach

2.1 VQ-GAN and VQ-VAE

当前我们仍旧处在设计阶段，为了在之后完整、具体的任务中搭建高质量的模型，我们需要首先对 GAN 与 VAE 有一些实践性的了解，于是我们分别对它们展开了研究。

2.1.1 GAN

我们首先研究的是 GAN 网络的搭建，以下是我们对 GAN 网络原理的认识：Generative Adversarial Network (GAN) 由两个部分组成，生成器（Generator）与辨别器（Discriminator）组成。

生成器的结构由编码器（Encoder）与解码器 (Decoder) 组成，其中在 Vector Quantized 意义下的 VQ-GAN 还包括一个编码器与解码器中间的 Codebook，也就是一个向量空间，每次编码器生成的编码都会去在向量空间中寻找最近邻，然后将最近邻传入解码器，在最基础的 GAN 架构中并没有这个环节。

解码器完成解码之后会生成一个图像，图像将会传入辨别器，辨别器判断图像是否为真，之后执行梯度反向传播。

用比喻来描述的话，GAN 网络是由一个画家与一个鉴赏师组成，画家从完全不会画画开始学习画画，每次画完画之后会让鉴赏师判断画是否为真画，同时，鉴赏师一开始也只是一个见习鉴赏师，在每一次画画与鉴赏的过程之中，他们都会分别获得对应的成长，直到最后已经专业的鉴赏师再也区分不清画家的画与真实的画，这时我们训练完成，可以撤掉鉴赏师，让画家去生成各种各样的画。

2.1.2 VAE

我们完成了如下架构的代码复刻，但是还没有开始展开训练，在这个复刻的过程中，我们领会到了训练的过程中最重要的是要构建一些梯度反向传播的链条，使得最终的损失函数的梯度反传可以从 decoder 传播到 encoder 与 codebook 部分，尤为关键的一点在于 encoder 与 codebook 的衔接部分涉及到 argmin 操作（用以找最近邻），argmin 并不具备梯度，于是我们学会了用 '$quantized = inputs + (quantized - inputs).detach()$' 的方式来保证梯度的顺利传播。

我们在当前的时间，对 VQ-GAN 与 VQ-VAE 的研究大概是如此，在之后有推进之后，我们将会决定究竟是使用 VQ-GAN 还是 VQ-VAE。

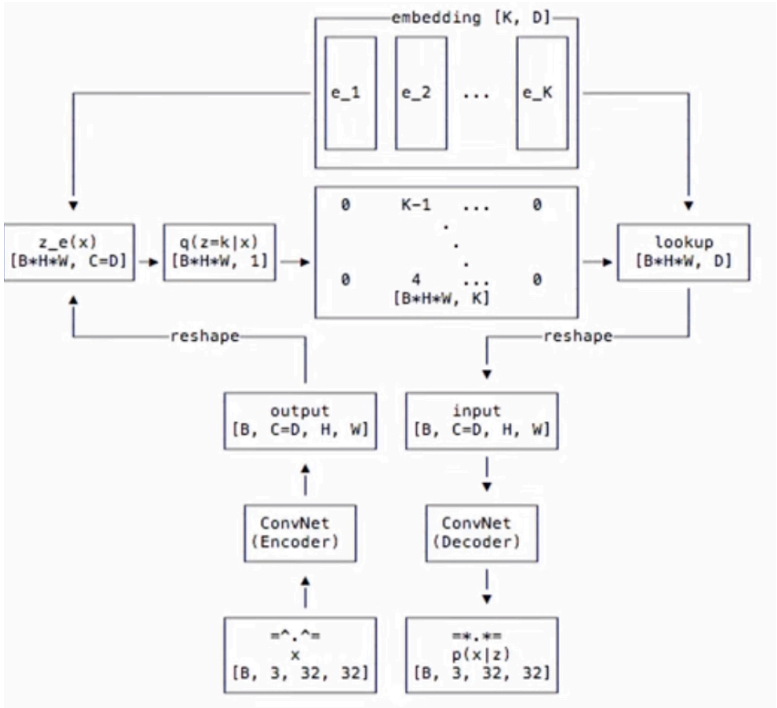

图 1: VQ-VAE 模型结构示意图

2.2 Bidirectional Transformer

Transformer 部分为 MaskGIT 性能大大提高的关键所在，其开创性地使用了 mask token 的概率作为输出，实现掩码分布调控以及后续图像的准确生成。传统的 transformer 由编码器 (Encoder) 和解码器 (Decoder) 两部分组成。而在该试验中，我们更倾向于把 transformer 的编码器和解码器部分拆开，实现一个 BERT (Bidirectional Encoder Representation from Transformers) 的结构。在该结构中，我们着重实现传统 transformer 的 Encoder 部分，如下图所示：

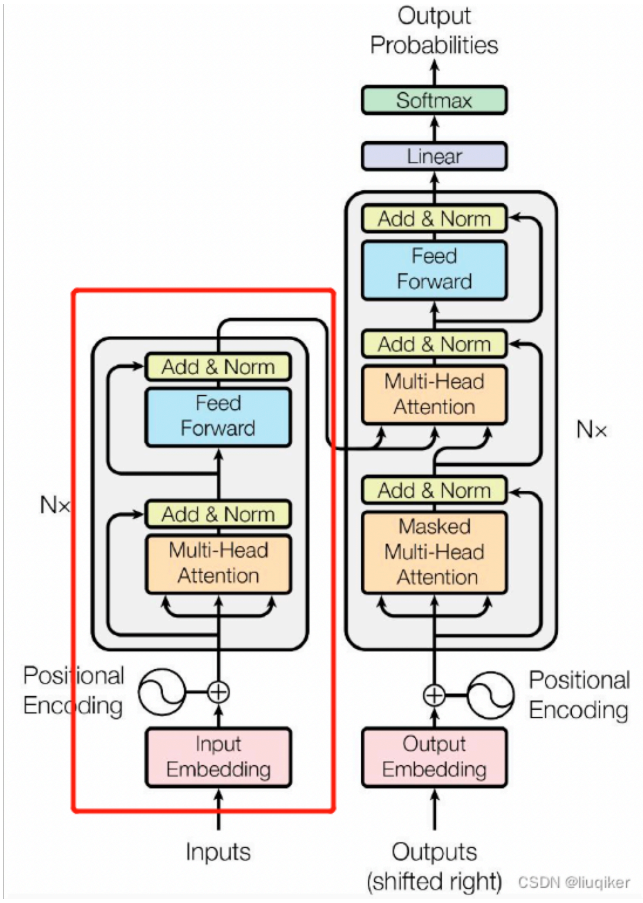

图 2: MaskGIT transformer 主体结构

根据这一思想我们设计了 transformer 的主体结构，即可分为生成嵌入向量、多头注意力以及前馈神经网络三大主要部分。这个给予 BERT 设计的双向 transformer 本质上以 visual token array 为输入，mask predict 为输出，方便后续解码器的解码和图像的生成。而目前 transformer 的模型架构也依照上述的三个部分来搭建。目前已经基于 torch.nn 完成了 bidirectional transformer 的搭建，但训练以及测试还在进行中。

3 Preliminary Results

我们使用最基础的 GAN 完成了在 MINIST 手写数据集上生成手写数字的测试任务。

我们可以简单的讲解一下实验过程, 结构如下:

①生成器由 MLP 组成，最终连接 '$Tanh()$' 激活函数将生成图像的数值映射到 $[-1, 1]$ 之间。

②辨别器也由 MLP 组成，最终连接 '$Sigmoid()$' 激活函数生成一个 $[0, 1]$ 之间的概率 p。

训练过程如下:

③训练的损失函数 '$loss\_fn = torch.nn.BCELoss()$' 用以进行 p 与目标期望（True or False）之间的差异刻画

④每次训练，向生成器中随机传入一些噪音，根据噪音生成一些图片，将这些图片送入辨别器,对于生成器我们有 '$g\_loss = loss\_fn(discriminator(pred_images.to(device)).cpu(), torch.ones(batch\_size, 1))$', 含义如下: 训练生成器时，我们期望辨别器将生成的照片预测为真; 对于辨别器我们有:'$d\_loss = loss\_fn(discriminator(gt\_images.to(device)).cpu(), torch.ones(batch\_size, 1)) + loss\_fn(discriminator(pred\_images.detach().to(device)).cpu(), torch.zeros(batch\_size, 1))$'。含义如下: 我们期望辨别器将生成的照片预测为假，真实的照片预测为真。

以下图片是我们生成的照片:

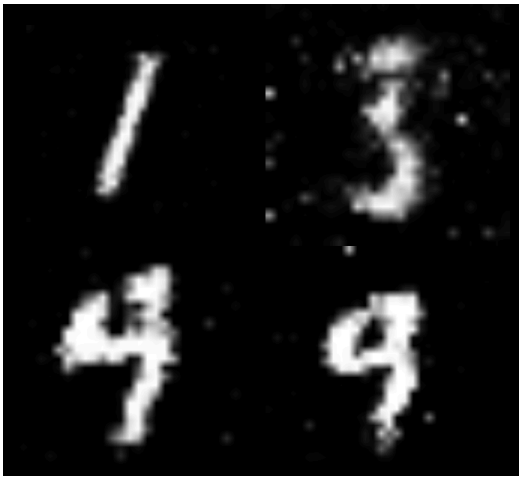

图 3: 根据 MINST 手写数据集生成的图片