Final Report

受到ZHB的影响, 这周我主要在看Transformer 架构的流模型搭建. 基于网上的资料, AI的解释和论文 *Attention is all you need* 我学习了Transformer的思想原理和实现.

下面是我这周学习内容的简要概括:

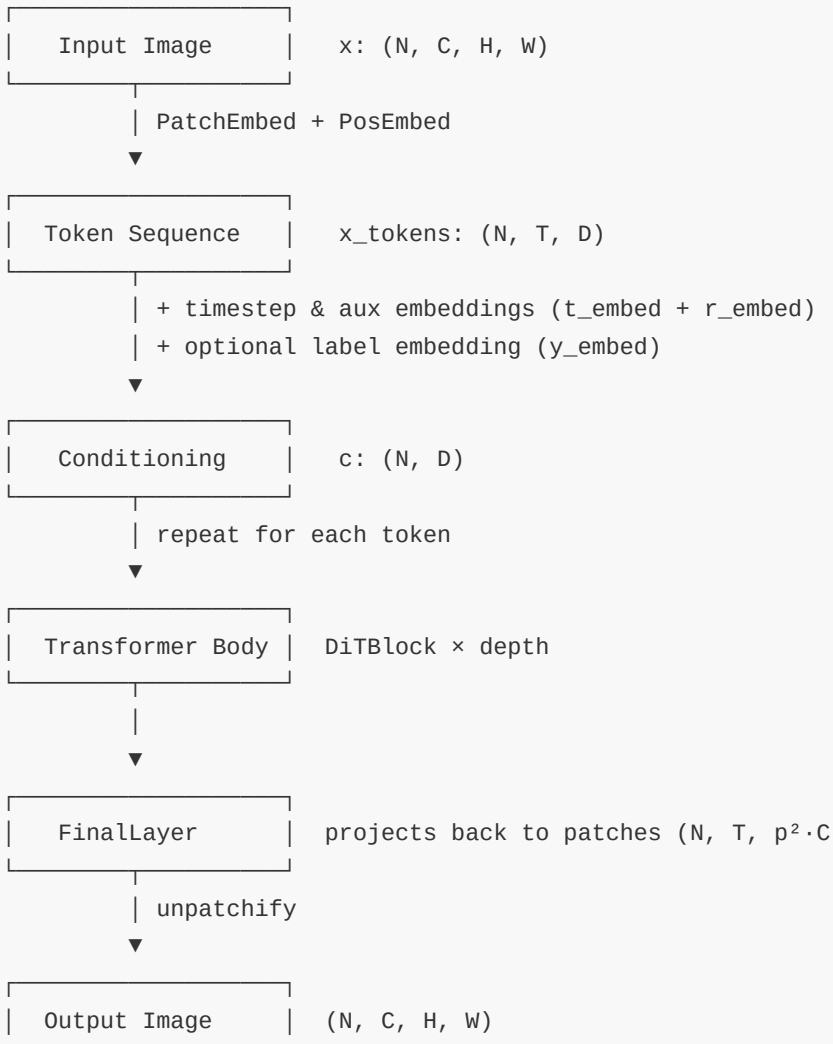
- 认识了原来处理离散序列信息(Sequence2Sequence)的Transformer的基本概念., 比如 token, key , value, 和query. 搞懂了常见的Transformer的架构中具体的数据运算. 比如, LayerNorm, AutoEncoder, MHSA 和 Feedforward layer的数学定义和矩阵运算.
- 从NLP的模型转到对ViT和DiT的学习. 我了解两者都是主要挪用了Transformer的Encoder, 把图片转化成Patches, 在用自注意力机制学习整张图片的信息, 最后在转化缝合成为原来的图片.
- 学习具体的代码实现. 我阅读了老师推荐的github代码:https://github.com/haidog-yaqub/MeanFlow 以及ZHB的project. 搞懂了具体每个部分的代码逻辑.
- 参考同学的项目和现有的代码, 自己搭建简单project, 并写了metric函数, 进行一些实验和测试, 进行了思考取得了一些对task的认识.我认为我基本完成了暑期科研项目

> 当然, 我在学习的过程中也越来越了解到, 机器学习领域非常宽广, 我不熟悉也不理解的东西还有很多.

接下来, 我主要聚焦与我对暑期科研项目代码的梳理和我的实验结果分析讨论:

我的代码不对图片进行autoencoder embedding. 也就是不把图片转化到latent space中.

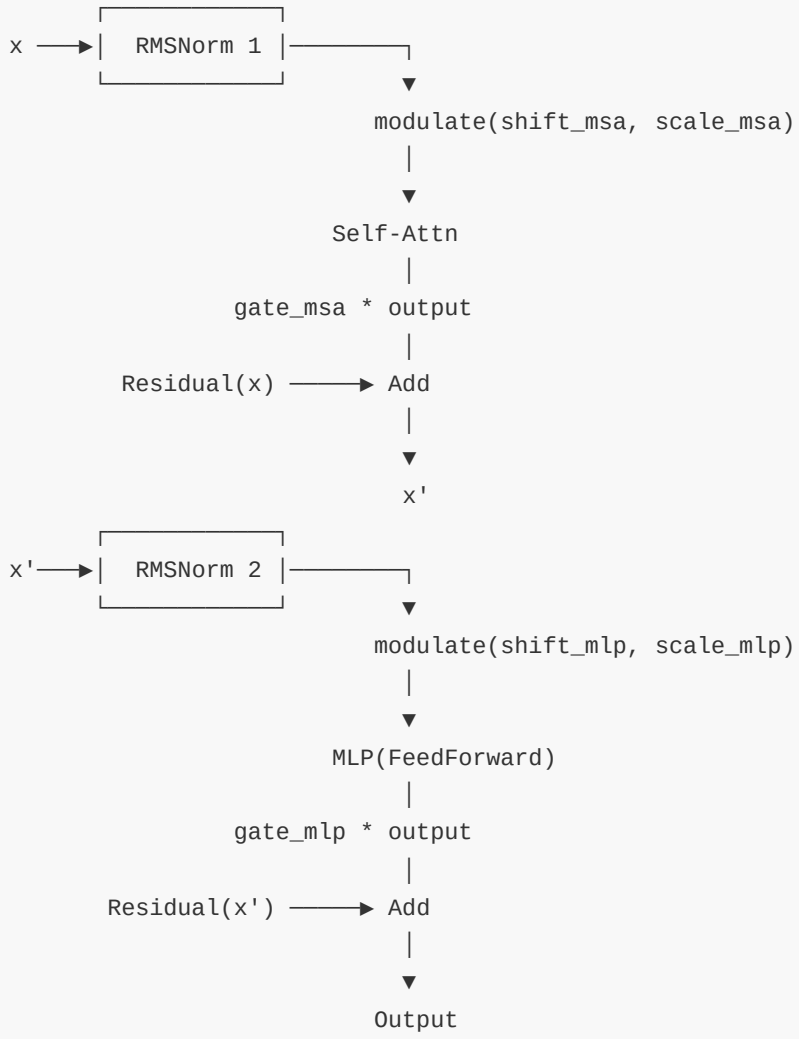
我整理的代码主要架构(modify from GPT):



```
    └──────────────────────┘
```

这是我针对最复杂的transformer body (DiTBlock)的整理:

```
  the coefficietn shift_msa, scale_msa and gate_msa, as well as shift_mlp
scale_mlp and gate_mlp is all generate from the output of
self.adaLN_modulation(c) which is nn.Sequential(nn.SiLU(), nn.Linear(dim, 6 *
dim)) a mlp to instill the influence of class infomation.
```



训练中meanflow中使用adaptive L2 loss, emphasis on easy sample, neglect the hard ones (可能是outliers/ too noisy), 优化器使用常用的AdamW

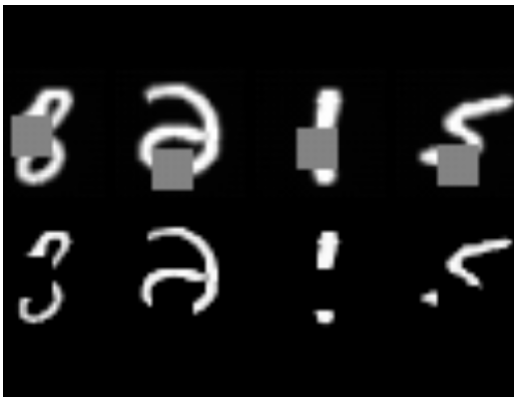
其他的比如生成 和保存checkpoints/ image略过.

evaluation: 使用loss/ mse_loss 对于训练程度做评估; 使用psnr和test dataset对于训练效果做评估.

最后, 我有一些疑问和自己的思考:

- image restorer是不需要class信息的, 也就是不需要cfg的修复图片, 我们希望model自己辨别和修复.
- 但是, 我们是否可以使用cfg训练出来的denoise model(input: pure gaussian noise, output: clean image)的参数? 这样, 就只需要微调就行了,大大地加快了模型的收敛速度. 训练过程中, 把class设置为uncond 的默认值, 其他保持不变, 让模型自主 train and do inference

- 由于ZHB只是做了(输入 x_noisy = eta ·gaussian_noise + clean_image ,;output: retored image) 我作了其他几组实验: (input: x_noisy = (1)eta·gaussian_noise + clean_image , (2)sp_value· saltpepper_noise + clean_image (3) block_mask + clean_image output: clear image) 我一共还做了单独了其他两种噪声+ 三种噪声混合.这是我对实验结果的思考.

> 对于掩膜修复和salt pepper restore组:
>
> model在一开始试图修复缺失的部分, 之后就开始cheat 走捷径: 直接用一平均灰度的方块填充被遮挡的区域.
>
> 如下是一张具体案例图:
>
> 
>
> 模型在初期修复sp noise能力比较优秀, 后期开始出现混乱.
>
> 如下是一张具体案例图:
>
> 
>
> 之后我上网查询后, 了解到, 我这种model还是对任务类型有比较强的限制的, 我使用pretrain model 参数本身针对从 纯gaussian noise 到 image的denoise 生成, 并不匹配 从x_noisy = sp_value·saltpepper_noise + clean_image 或者 block_mask + clean_image output: clear image 的 修复任务. 我意识到, 明确模型处理的任务是很重要的.
>
> 除此之外, 由于我采用了L2 loss, 针对全局pixel感知, 所以对与block mask修复能力弱, 需要引入比较复杂的感知损失和对抗性损失, 针对context consistency 做要求.

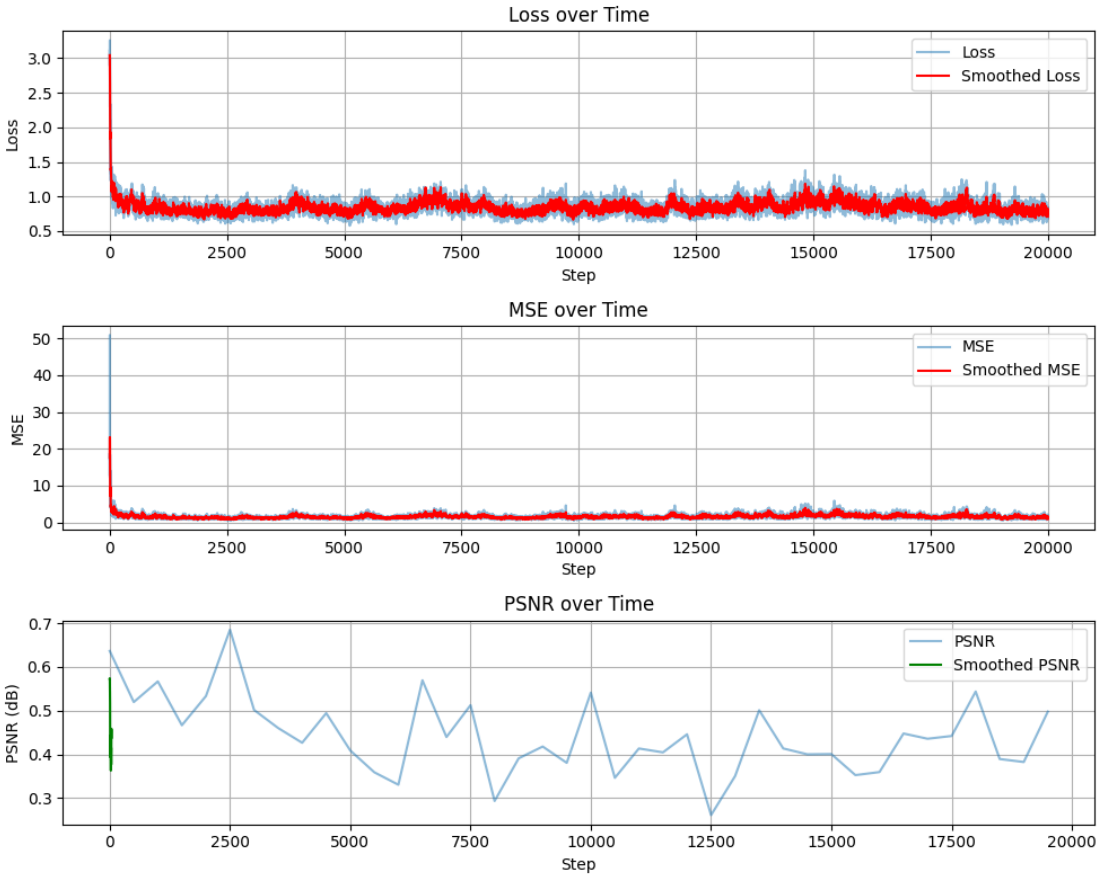
- 我之后了解到, 还有gaussian blur , 就是用模糊核卷积图像, 然后训练model deblur回到sharp image.但是我没有做了.

之后是对我训练的model分析:

> 也许是我计算机GPU显存太小, 使得我的模型参数少(加起来只有2.5MB大小), 总体修复能力尚可, 但是会犯错:
>
> 这是一个案例图:

在这张图片中, 0,9 都有较好的修复, 但是,模型却把5 修复成了6.

loss, PSNR 的结果:



我的代码放在:https://github.com/flashfire1001/image-Restorer

(整合了unet, dit全放在models, train.py用于训练模型, meanflow是进行meanflow训练的loss 和sample的函数, demo是load checkpoint生成一张图片展示模型效果.train_latent没有使用.)

我认为我的暑期科研以及基本完成, 请老师看看我还有什么地方可以延伸学习, 或者哪里做的还不够充分.