深度学习第三次作业

姜春飞——2021214053

Part One： RNN and Transformer

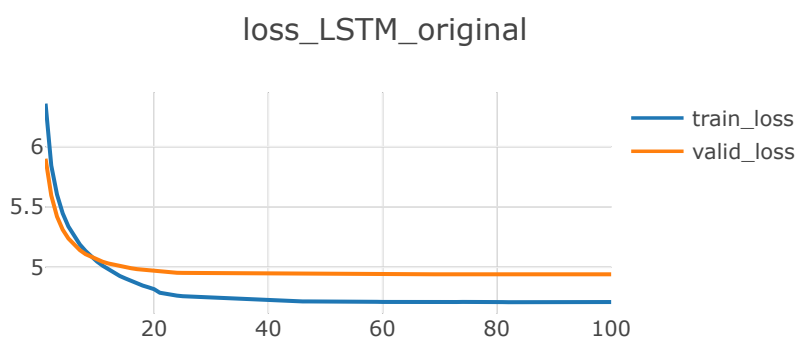
Task A：LSTM

基于 pytorch 实现了 LSTM 语言模型，并使用 Adam 优化器。

超参数设定：

超参数名	超参数值
epoch	100
train_batch_size	20
eval_batch_size	10
max_sql	35
词嵌入的向量维度	200
hidden state 的向量维度	200
LSTM 层数	2
dropout	0.5

LSTM 在训练集和测试集上的 Loss 曲线：



验证集最优 Loss：4.941922

测试集最优 Loss：4.711423

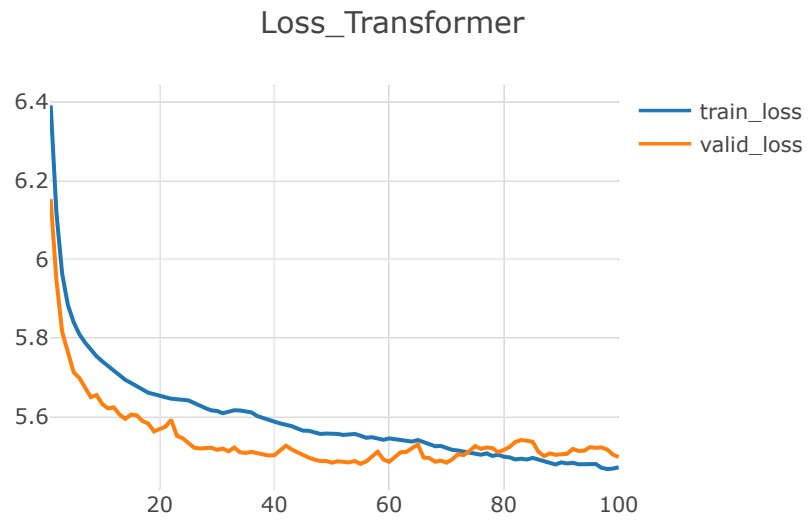
Task B：Transformer

基于 pytorch 实现了层数为 2 的 Transformer 语言模型，并使用 Adam 优化器。

超参数设定：

超参数名	超参数值
epoch	100
train_batch_size	20
eval_batch_size	10
max_sql	35
词嵌入的向量维度	200
hidden state 的向量维度	200
Transformer 层数	2
dropout	0.5

Transformer 在训练集和测试集上的 Loss 曲线：



验证集最优 Loss：5.486737

测试集最优 Loss：5.467322

Data Preparation

数据处理阶段对文本数据的处理和对图像数据的处理有很大的不同，其中：

对于文本数据的预处理：

分词：对文本数据进行分词。

建立字典：在分词后对单词进行序列化，即为每个单词赋予一个唯一的 id，生成一个 vocabulary。

将数据集中的文本和单词转化为对应的索引或 id 序列。

对于图像数据的预处理：

几何：平移，旋转，剪切等对图像几何改变的方法，对模型的泛化能力有所增强。

色彩：主要是亮度变换，如使用 HSV(HueSaturationValue)增强。

随机擦除：主要是模拟遮挡，从而提高模型泛化能力，对遮挡有更好的鲁棒性。

Technical Details

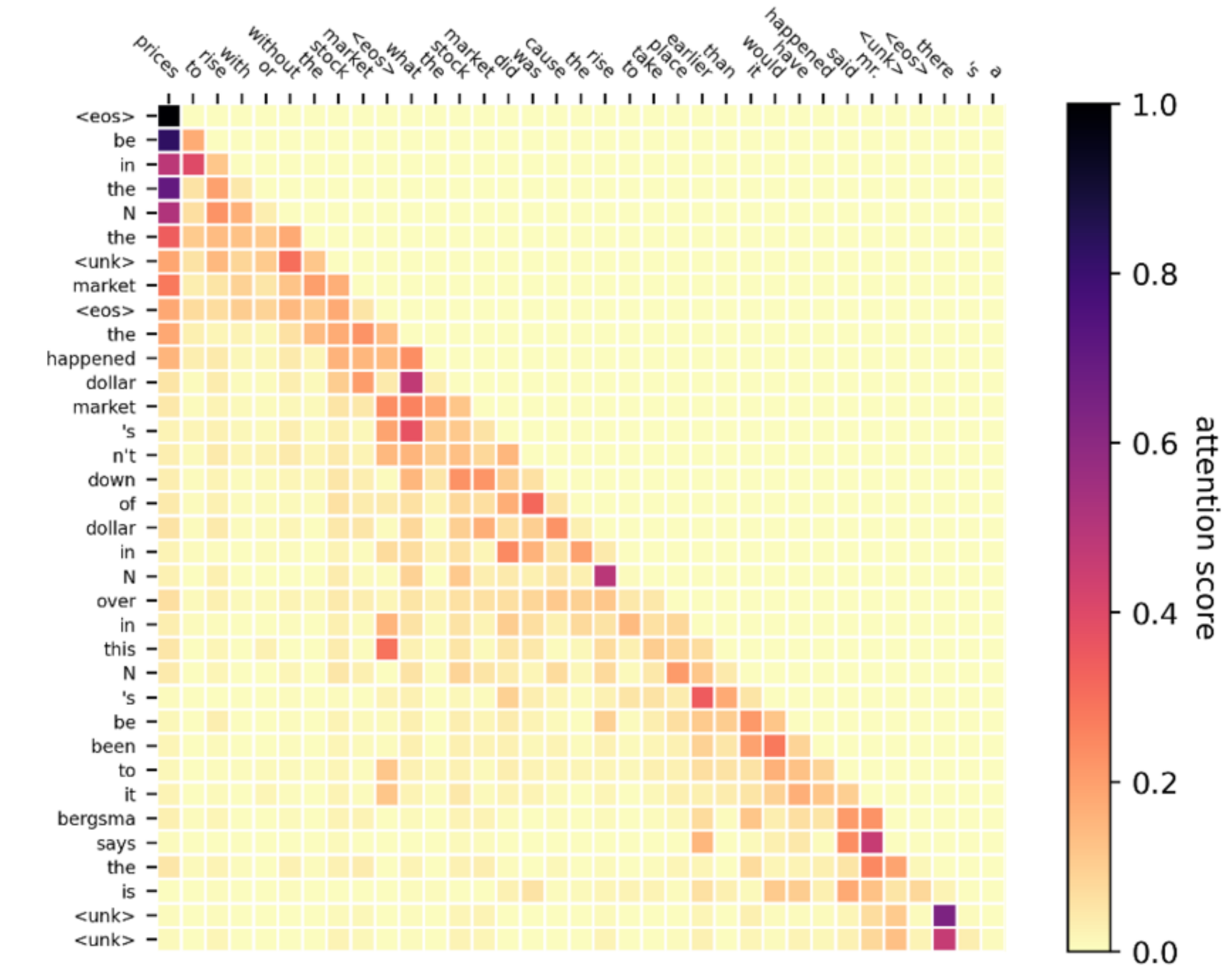
mask 在 Transformer 中的必要性：

在 encoder 和 decoder 两个模块里都有 padding mask，位置是在 softmax 之前。由于 encoder 和 decoder 两个模块都会有各自相应的输入，但是输入的句子长度是不一样的，计算 attention score 会出现偏差，为了保证句子的长度一样所以需要进行填充，但是用 0 填充的位置的信息是完全没有意义的，经过 softmax 操作也会有对应的输出，会影响全局概率值，因此我们希望这个位置不参与后期的反向传播过程。以此避免最后影响模型自身的效果，既在训练时将补全的位置给 Mask 掉，也就是在这些位置上补一些无穷小的值，经过 softmax 操作，这些值就成了 0，就不再影响全局概率的预测。

mask 在代码中的实现：

```
def generate_square_subsequent_mask(self, sz):
    mask = (torch.triu(torch.ones(sz, sz)) == 1).transpose(0, 1)
    mask = mask.float().masked_fill(mask == 0, float('-inf')).masked_fill(mask == 1, float(0.0))
    return mask
```

Attention Visualization



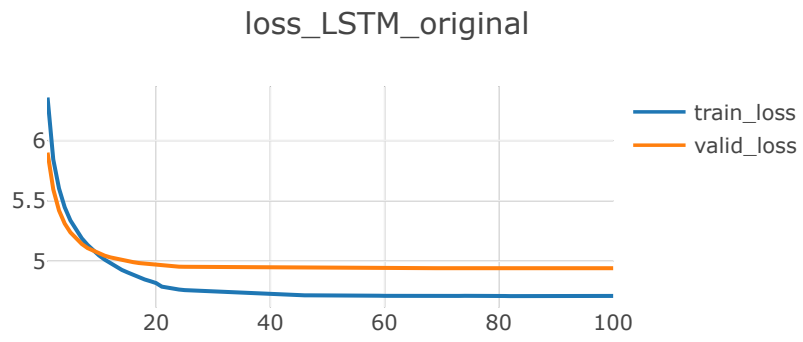
从可视化结果可以看出，模型正确预测出"stock"的下一个单词为 market，证明模型可以学习到一定的意义。

Extra Techniques

LSTM 性能提升：

根据 Lior Wolf 等人在 2016 年提出的理论[1]，在训练语言模型时，绑定输入嵌入和输出嵌入(共享嵌入矩阵, weight tying 能够提升语言模型的表现。使用这种方案对 LSTM 进行了改进，改进效果如下：

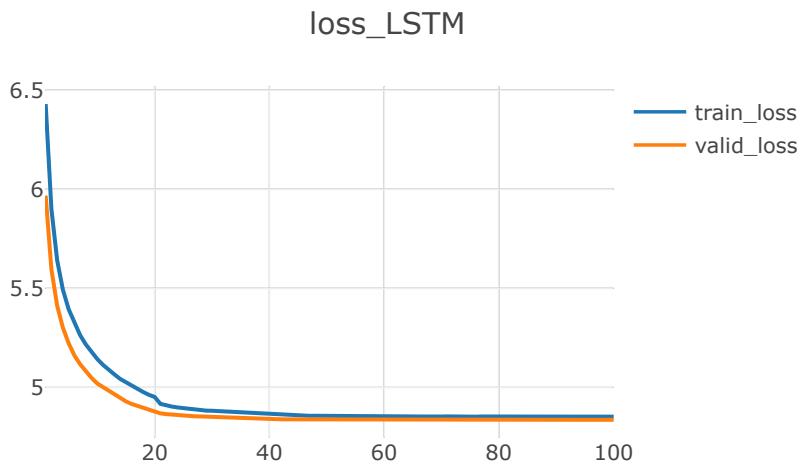
未改进版本：



验证集最优 Loss：4.941922

测试集最优 Loss：4.711423

改进版本：



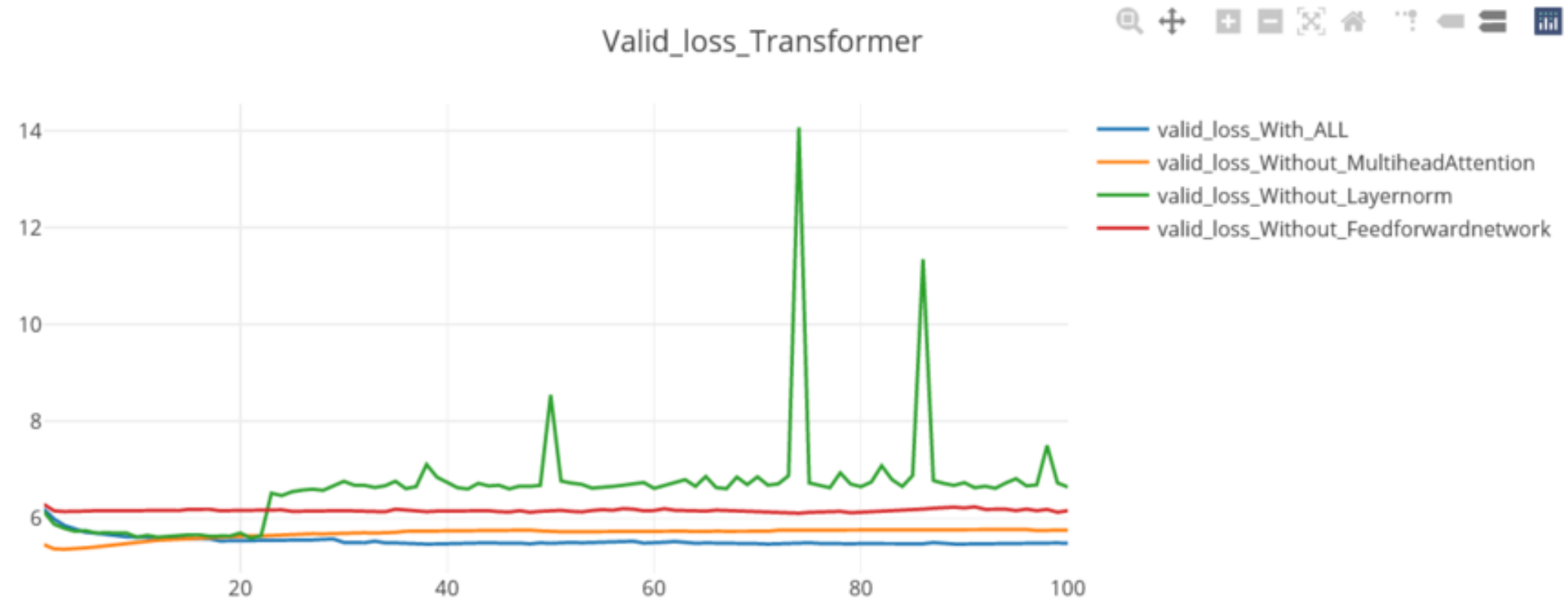
验证集最优 Loss：4.836603

测试集最优 Loss：4.853021

从实验结果也可以看出，使用如上理论可以一定程度提升 LSTM 的表现。

消融实验：

分别将 Transformer 之中的 multi-head attention, feed-forward network, Layernorm 去除，得到以下结果：

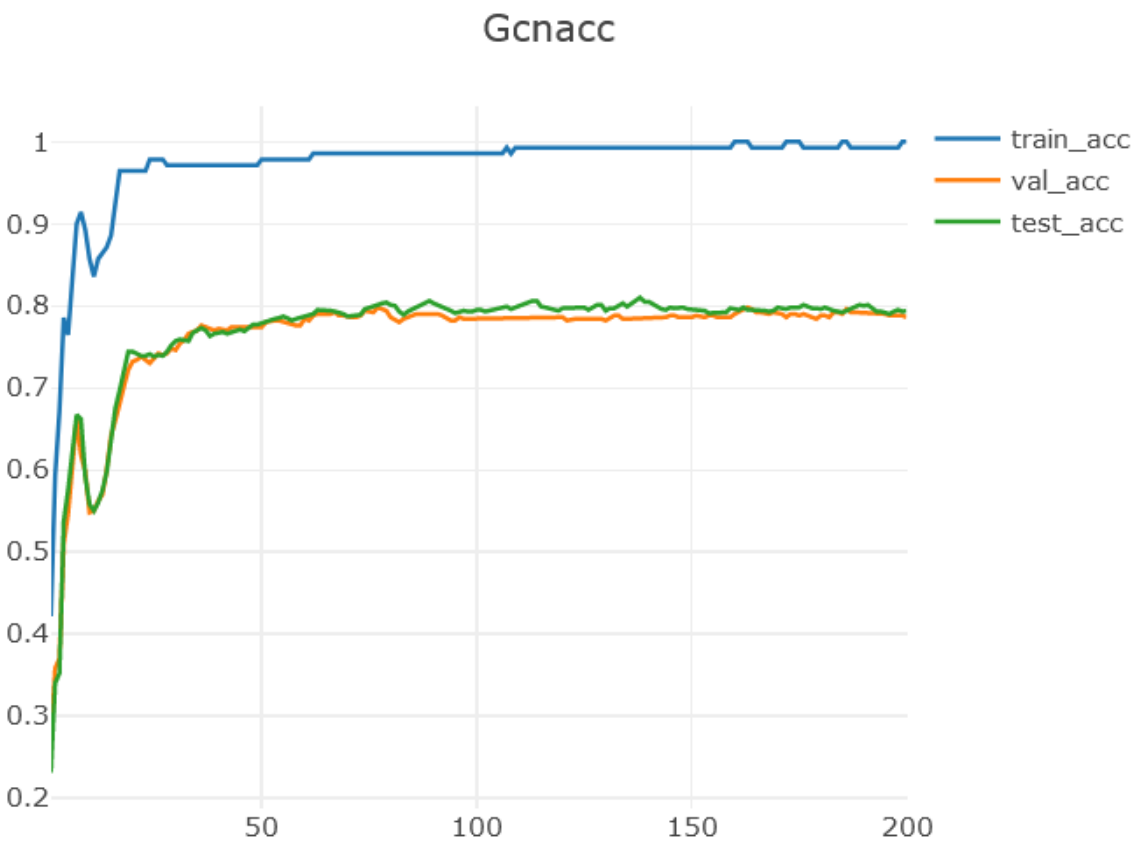


从实验结果之中可以看出， Layernorm 对模型的影响最大，去除 Layernorm 使得训练不收敛。去除 feed-forward network 模块对模型的影响较大。而去除 multi-head attention 对模型的影响最小。

Part Two: GNN and Node2Vec

Task A:

GCN

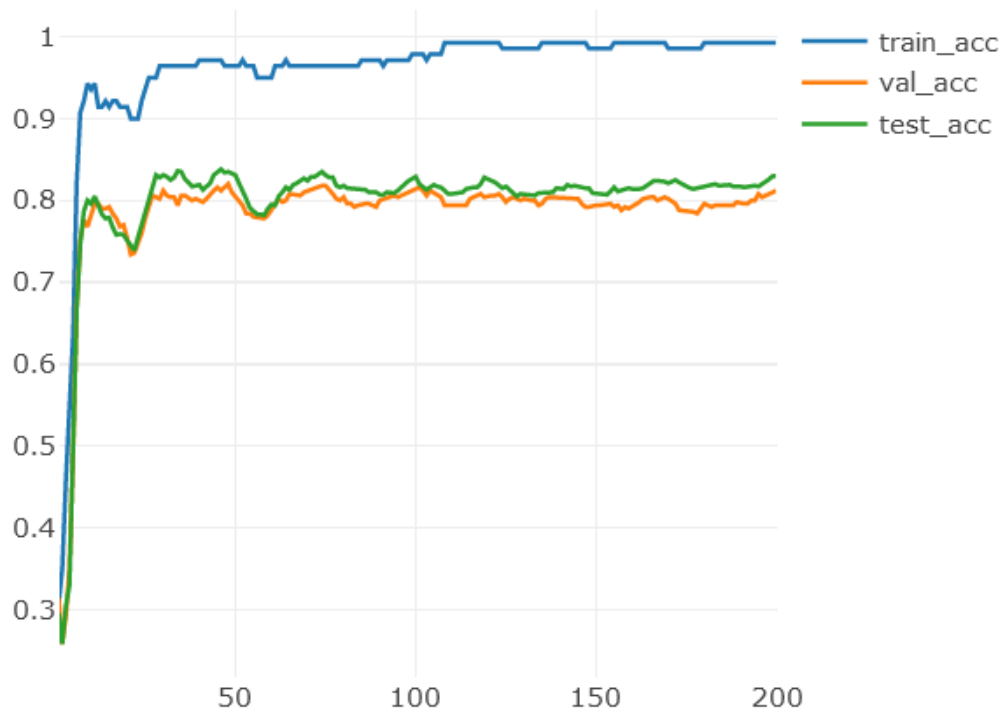


测试集最优精度：0.804

验证集最优精度：0.821

最佳 epoch：113

GAT

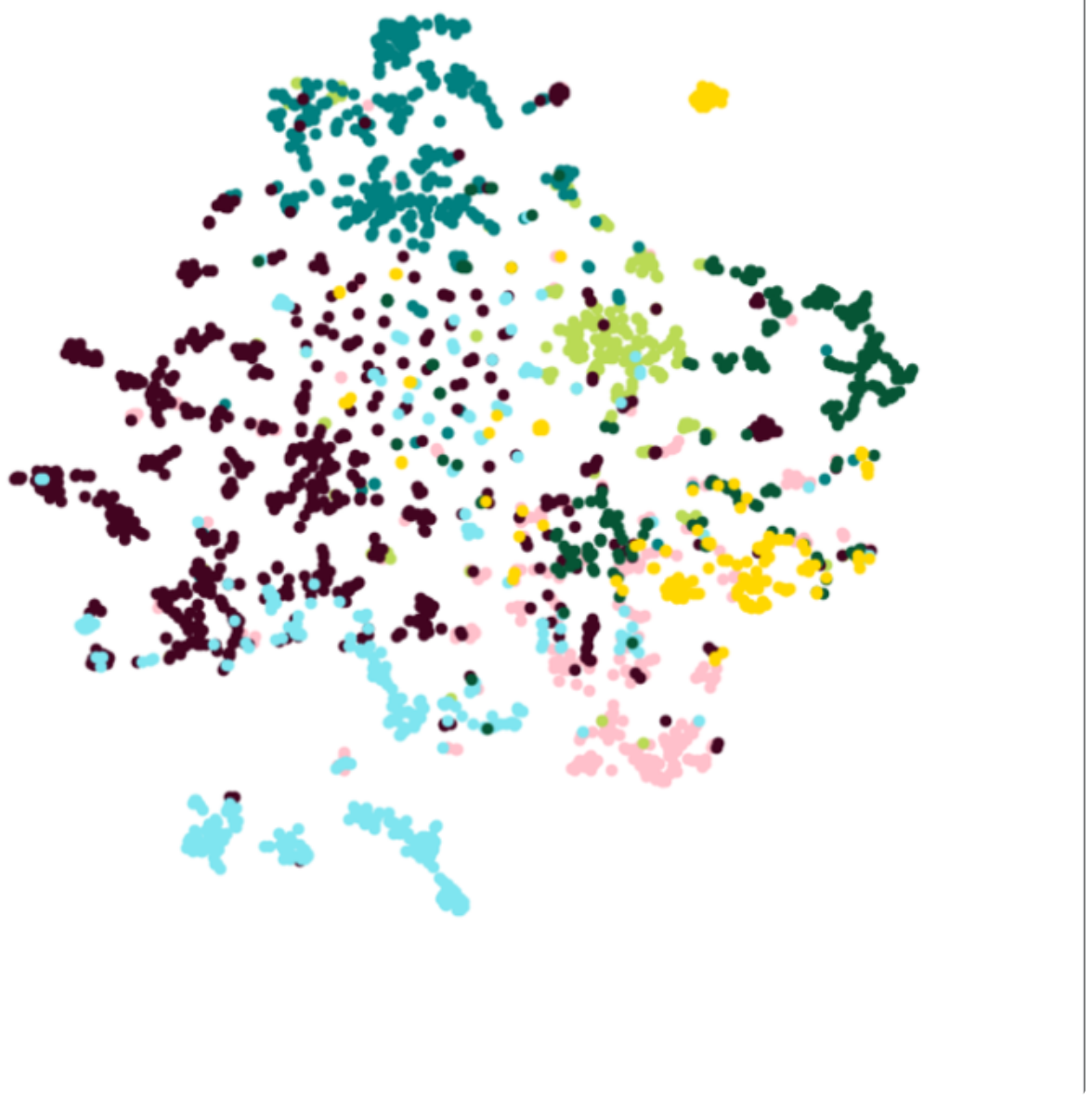


测试集最优精度：0.831

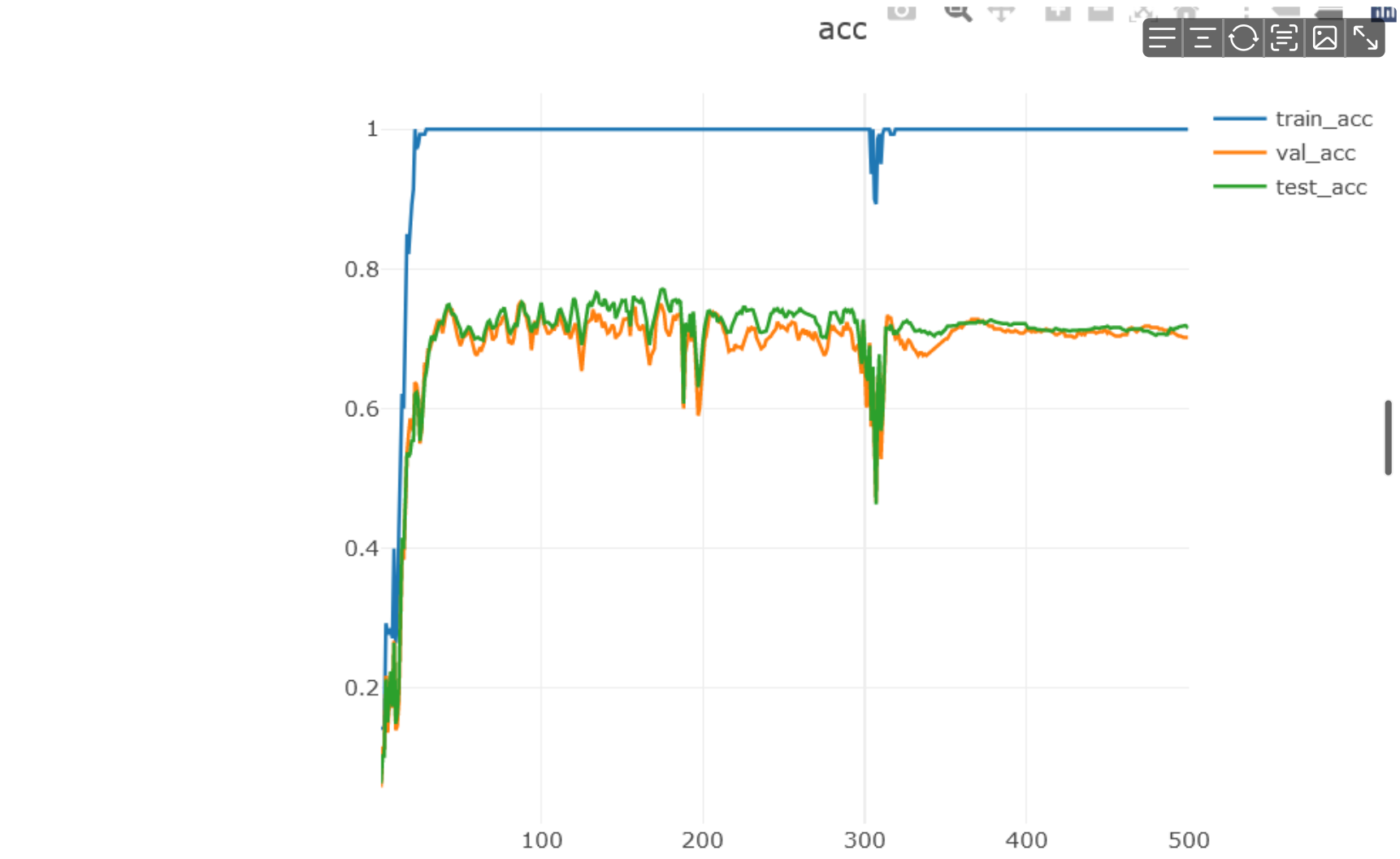
验证集最优精度：0.816

最佳 epoch：164

Node2vec



Task B：DEEPGCN



测试集最优精度：0.77

验证集最优精度：0.744

最佳 epoch：132

参考文献：

[1] Press O , Wolf L . Using the Output Embedding to Improve Language Models[J]. 2016.