**Link to my forked repositories:**

**https://github.com/ZonaWei**

Name: Mingrou Wei

Matriculation number: 23-739-436

I used CPU to train the model and modify the ymal file by adding ‘layer\_norm: "pre"’ to change the default postnorm setting to prenorm setting. The command line to train both model is ‘mt-exercise-03 % bash ./scripts/train\_prenorm.sh’. All the data is stored in ‘validation.txt’ of both models.

表格

AI 生成的内容可能不正确。

（Table containing the validation perplexities of the three models）

图表, 折线图

AI 生成的内容可能不正确。

**Why does training progress differ?**

These differences are mainly due to how layer normalization interacts with gradient flow. In **Pre-Norm**, normalization is applied *before* the sublayers (attention and feed-forward), which helps stabilize the training signal early on. This preemptive normalization ensures that gradients do not vanish or explode, leading to faster and more stable convergence. In contrast, **Post-Norm** normalizes *after* the sublayers, which can cause instability during the initial training phase—particularly in deeper networks or when using larger learning rates. This explains why Postnorm performs worse under our current settings.

**How does our setup differ from Wang et al. (2019)?**

Wang et al. (2019) used deeper Transformer models (e.g., 6 or 12 layers) and much larger datasets such as WMT14, allowing for longer and more stable training. Their findings showed that **Post-Norm fails to train deep models effectively**, while Pre-Norm remains robust. In contrast, our setup is explicitly low-resource:

* Only **4 encoder** and **1 decoder** layer
* Just **100k sentence pairs**
* Relatively short training (under 10k steps)

These constraints **amplify early training behaviors**, which favors Pre-Norm even more and may exaggerate the differences. Additionally, unlike Wang et al., we do not compare BLEU scores or final convergence behavior in high-resource settings.