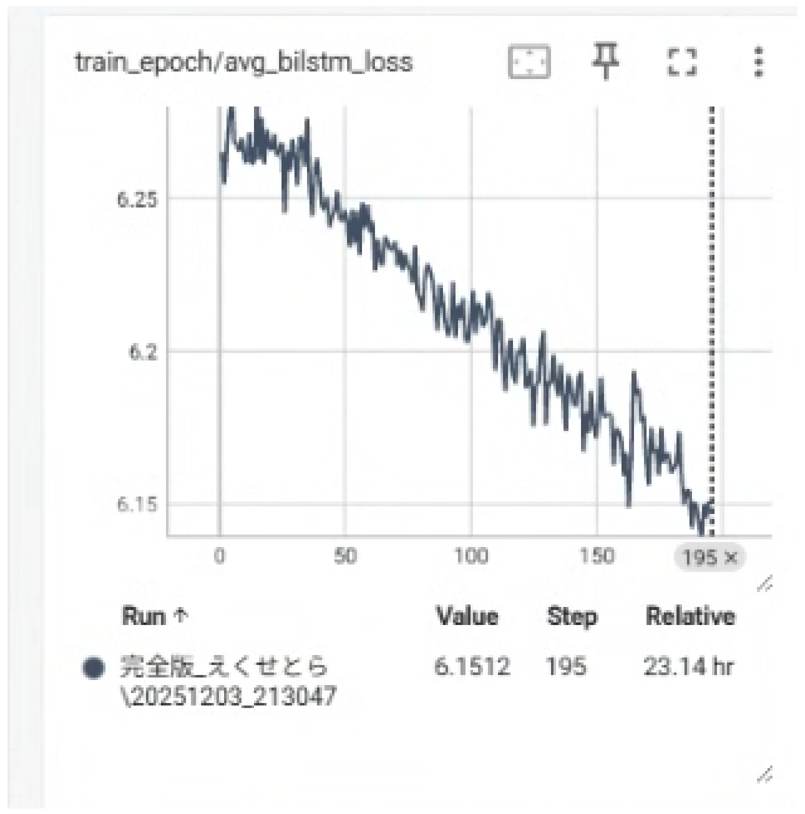
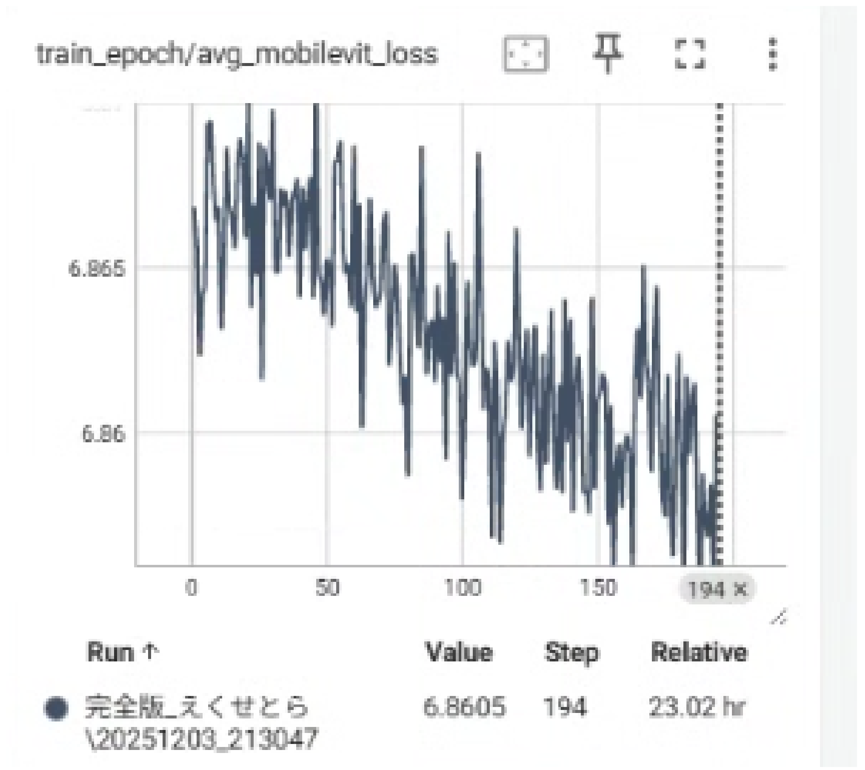
卒研にむけて②

提案手法前確認

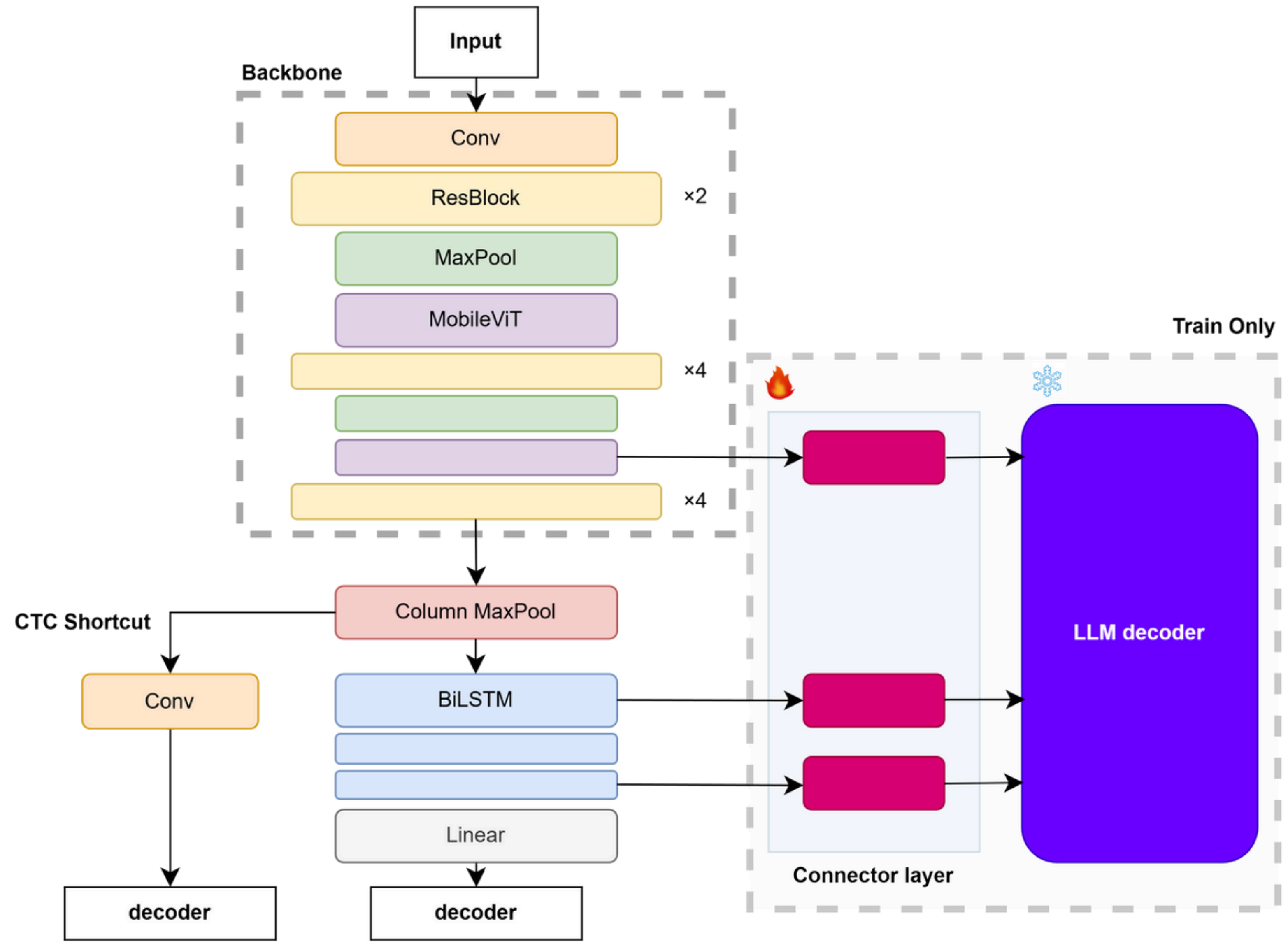


Run ↑	Value	Step	Relative
● 完全版_えくせとら \20251203_213047	6.1512	195	23.14 hr



Run ↑	Value	Step	Relative
● 完全版_えくせとら \20251203_213047	6.8605	194	23.02 hr

lossの比較



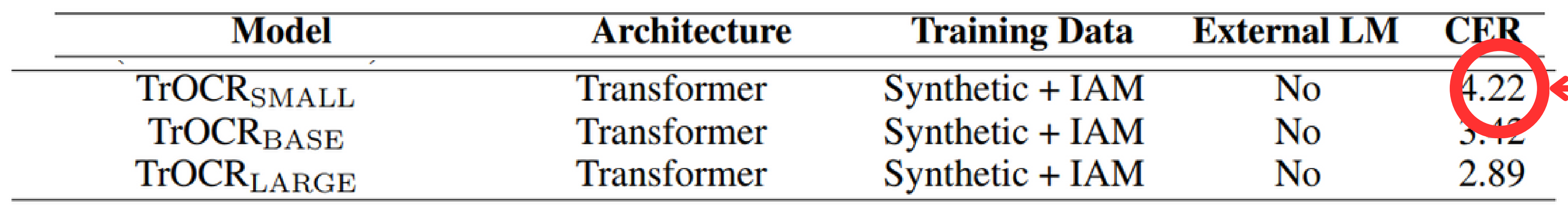
前のモデル

なにをしているのか

正しくファインチューニングできるのか（自分の工夫含まず）
→ファインチューニングできたら自分で書いた学習ループが正しいことが示せる

上記が確認できたら自分の工夫入れていこうと思う

Trocrファインチューニング編

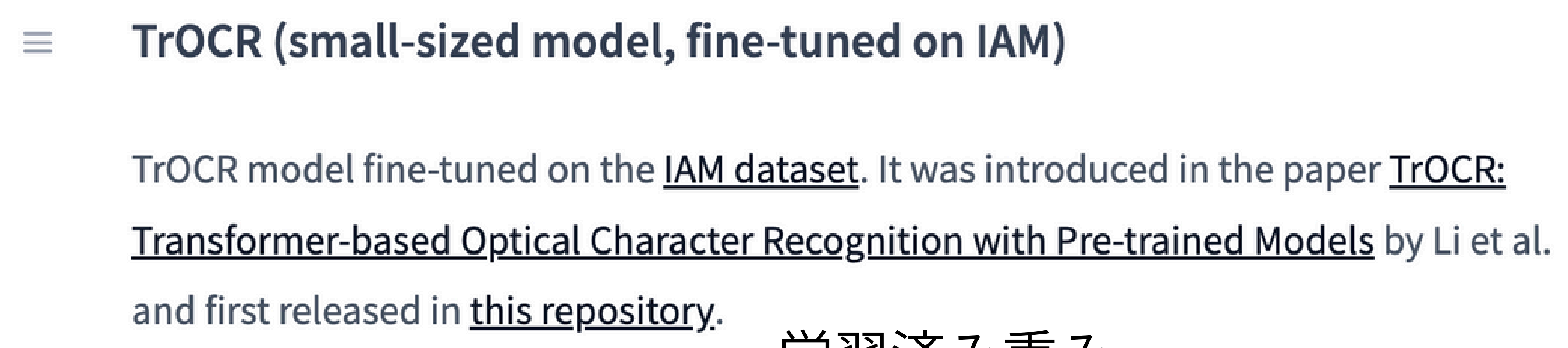


Model	Architecture	Training Data	External LM	CER
$\text{TrOCR}_{\text{SMALL}}$	Transformer	Synthetic + IAM	No	4.22
$\text{TrOCR}_{\text{BASE}}$	Transformer	Synthetic + IAM	No	3.42
$\text{TrOCR}_{\text{LARGE}}$	Transformer	Synthetic + IAM	No	2.89

目標値

smallで精度確認→CER：70超

↓



TrOCR (small-sized model, fine-tuned on IAM)

TrOCR model fine-tuned on the IAM dataset. It was introduced in the paper TrOCR: Transformer-based Optical Character Recognition with Pre-trained Models by Li et al. and first released in this repository.

学習済み重み

もうこれはすでに
IAMでファインチューニング済
なので同じデータとモデルを使っ
てるおりかつ正しいモデル設定、
データ前処理がなされていたら精
度がでるはず

Trocrファインチューニング編

モデル設定見直し

```
model.config.pad_token_id = 1
model.config.eos_token_id = 2
model.config.decoder_start_token_id = 2
```

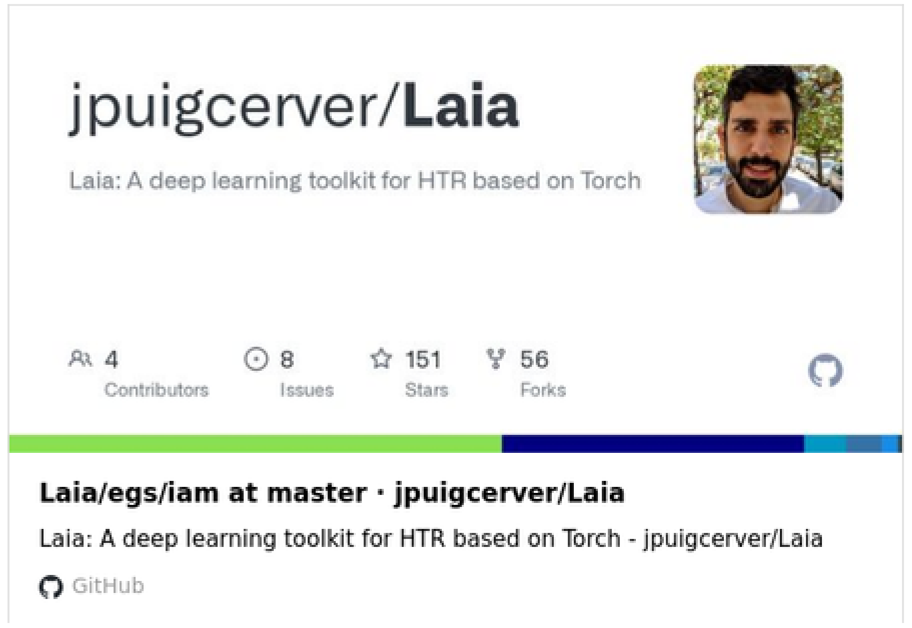
←CER:5.38

上記＋データ見直し

- train,valのデータ分布
- リサイズ
- 論文記載のデータ処理を使用

←CER:4.33

(目標値にほぼ到達で提案手法に移れる！！)



jpuigcerver/**Laia**

Laia: A deep learning toolkit for HTR based on Torch

4 Contributors 8 Issues 151 Stars 56 Forks

Laia/egs/iam at master · jpuigcerver/Laia

Laia: A deep learning toolkit for HTR based on Torch - jpuigcerver/Laia

GitHub

Trocrファインチューニング編

モデル設定見直し

```
model.config.pad_token_id = 1
model.config.eos_token_id = 2
model.config.decoder_start_token_id = 2
```

←CER:5.38

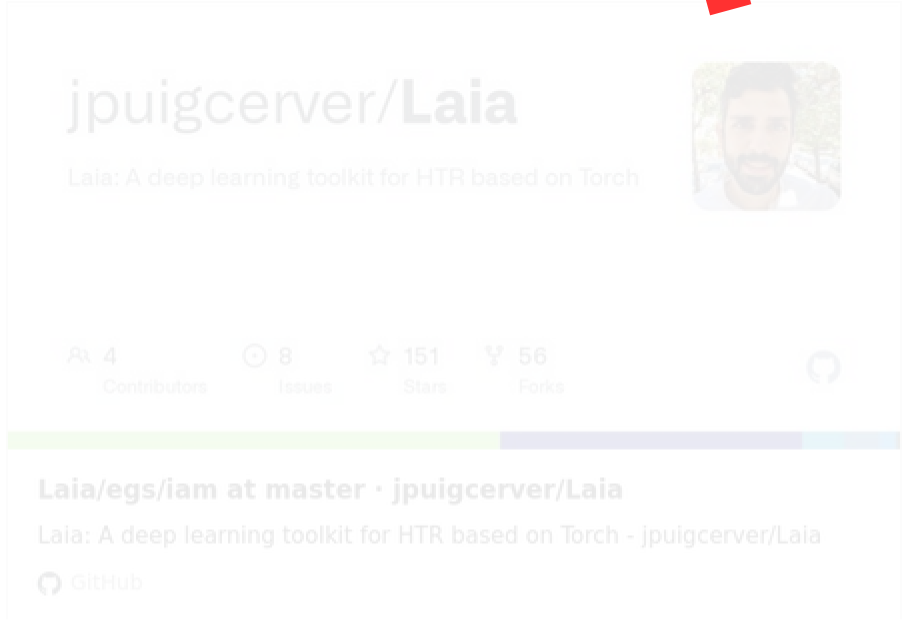
問題発生！！

データ見直し

- train,valのデータ分布
- リサイズ
- 論文記載のデータ処理を使用

←CER:4.33

(目標値にほぼ到達で提案手法に移れる！！)



jpuigcerver/Laia

Laia: A deep learning toolkit for HTR based on Torch

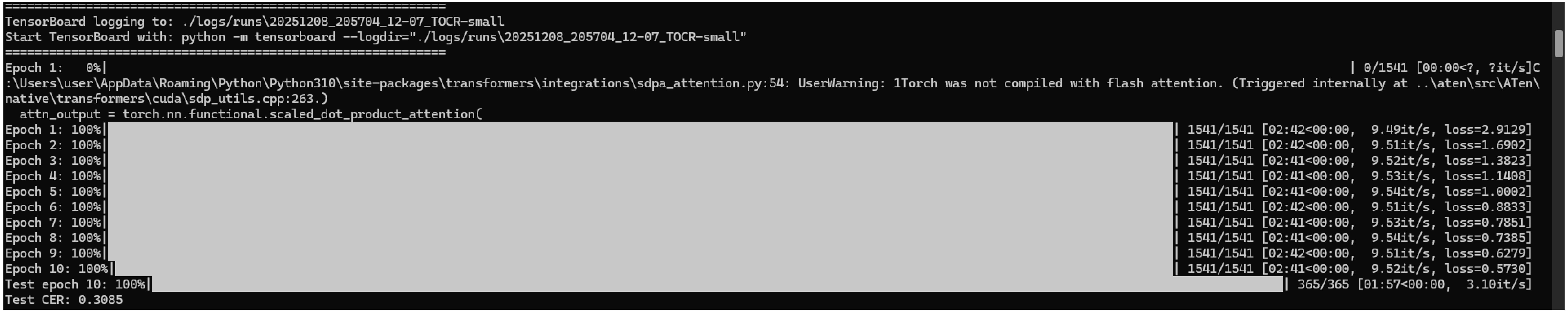
4 Contributors · 8 Issues · 151 Stars · 56 Forks

Laia/egs/iam at master · jpuigcerver/Laia

Laia: A deep learning toolkit for HTR based on Torch - jpuigcerver/Laia

GitHub

Trocrファインチューニング編



```
============================================================
TensorBoard logging to: ./logs/runs\20251208_205704_12-07_TOCR-small
Start TensorBoard with: python -m tensorboard --logdir="./logs/runs\20251208_205704_12-07_TOCR-small"
============================================================
Epoch 1:   0%|                                    | 0/1541 [00:00<?, ?it/s]C
:\Users\user\AppData\Roaming\Python\Python310\site-packages\transformers\integrations\sdpa_attention.py:54: UserWarning: 1Torch was not compiled with flash attention. (Triggered internally at ..\aten\src\ATen\
native\transformers\cuda\sdp_utils.cpp:263.)
  attn_output = torch.nn.functional.scaled_dot_product_attention(
Epoch 1: 100%|| 1541/1541 [02:42<00:00,  9.49it/s, loss=2.9129]
Epoch 2: 100%|| 1541/1541 [02:42<00:00,  9.51it/s, loss=1.6902]
Epoch 3: 100%|| 1541/1541 [02:41<00:00,  9.52it/s, loss=1.3823]
Epoch 4: 100%|| 1541/1541 [02:41<00:00,  9.53it/s, loss=1.1408]
Epoch 5: 100%|| 1541/1541 [02:41<00:00,  9.54it/s, loss=1.0002]
Epoch 6: 100%|| 1541/1541 [02:42<00:00,  9.51it/s, loss=0.8833]
Epoch 7: 100%|| 1541/1541 [02:41<00:00,  9.53it/s, loss=0.7851]
Epoch 8: 100%|| 1541/1541 [02:41<00:00,  9.54it/s, loss=0.7385]
Epoch 9: 100%|| 1541/1541 [02:42<00:00,  9.51it/s, loss=0.6279]
Epoch 10: 100%|| 1541/1541 [02:41<00:00,  9.52it/s, loss=0.5730]
Test epoch 10: 100%|| 365/365 [01:57<00:00,  3.10it/s]
Test CER: 0.3085
```

ログ

Trocrファインチューニング編

$$\text{loss} = -\frac{1}{|\Omega|} \sum_{(b,t)\in\Omega} \log p_\theta(y_{b,t} \mid \text{画像}, \text{文の過去トークン})$$

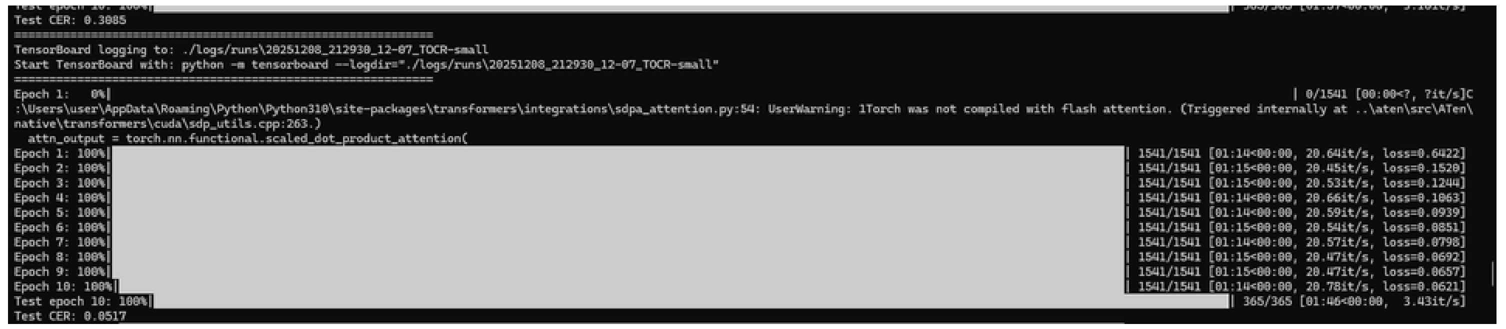
課題

学習を回すと精度が悪化する

- 現状：epoch:10、CER:　30.85
- 違和感：ロスは減少している、まず、valとtestで差が10倍

試したこと

- 1部の層のみ学習可能にする（encoderの最終層とdecoder２層）　→　CER：5.90
- ロスを自作（内部で何してるかわからん！）　→　CER ：8.06
- 上二つを組み合わせた：CER：5.16

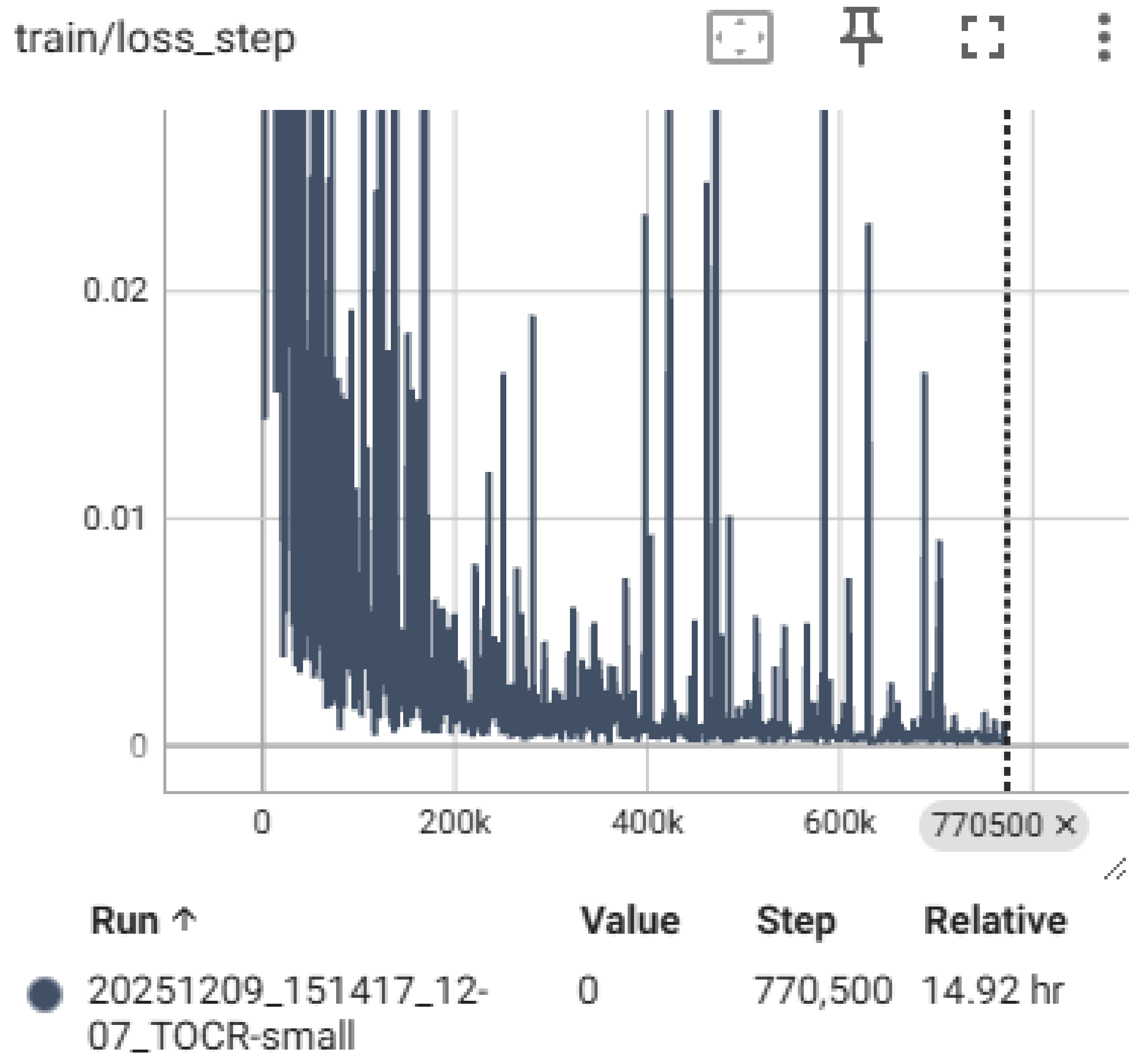


```
Test CER: 0.3085
==============================================================
TensorBoard logging to: ./logs/runs\20251208_212930_12-07_TOCR-small
Start TensorBoard with: python -m tensorboard --logdir="./logs/runs\20251208_212930_12-07_TOCR-small"
==============================================================
Epoch 1:   0%|                                                                      | 0/1541 [00:00<?, ?it/s]C
:\Users\user\AppData\Roaming\Python\Python310\site-packages\transformers\integrations\sdpa_attention.py:54: UserWarning: 1Torch was not compiled with flash attention. (Triggered internally at ..\aten\src\ATen\
native\transformers\cuda\sdp_utils.cpp:263.)
  attn_output = torch.nn.functional.scaled_dot_product_attention(
Epoch 1: 100%|                                                    | 1541/1541 [01:14<00:00, 20.64it/s, loss=0.6422]
Epoch 2: 100%|                                                    | 1541/1541 [01:15<00:00, 20.45it/s, loss=0.1520]
Epoch 3: 100%|                                                    | 1541/1541 [01:15<00:00, 20.53it/s, loss=0.1244]
Epoch 4: 100%|                                                    | 1541/1541 [01:14<00:00, 20.66it/s, loss=0.1063]
Epoch 5: 100%|                                                    | 1541/1541 [01:14<00:00, 20.59it/s, loss=0.0939]
Epoch 6: 100%|                                                    | 1541/1541 [01:15<00:00, 20.54it/s, loss=0.0851]
Epoch 7: 100%|                                                    | 1541/1541 [01:14<00:00, 20.57it/s, loss=0.0798]
Epoch 8: 100%|                                                    | 1541/1541 [01:15<00:00, 20.47it/s, loss=0.0692]
Epoch 9: 100%|                                                    | 1541/1541 [01:15<00:00, 20.47it/s, loss=0.0657]
Epoch 10: 100%|                                                   | 1541/1541 [01:14<00:00, 20.78it/s, loss=0.0621]
Test epoch 10: 100%|                                                      | 365/365 [01:46<00:00,  3.43it/s]
Test CER: 0.0517
```

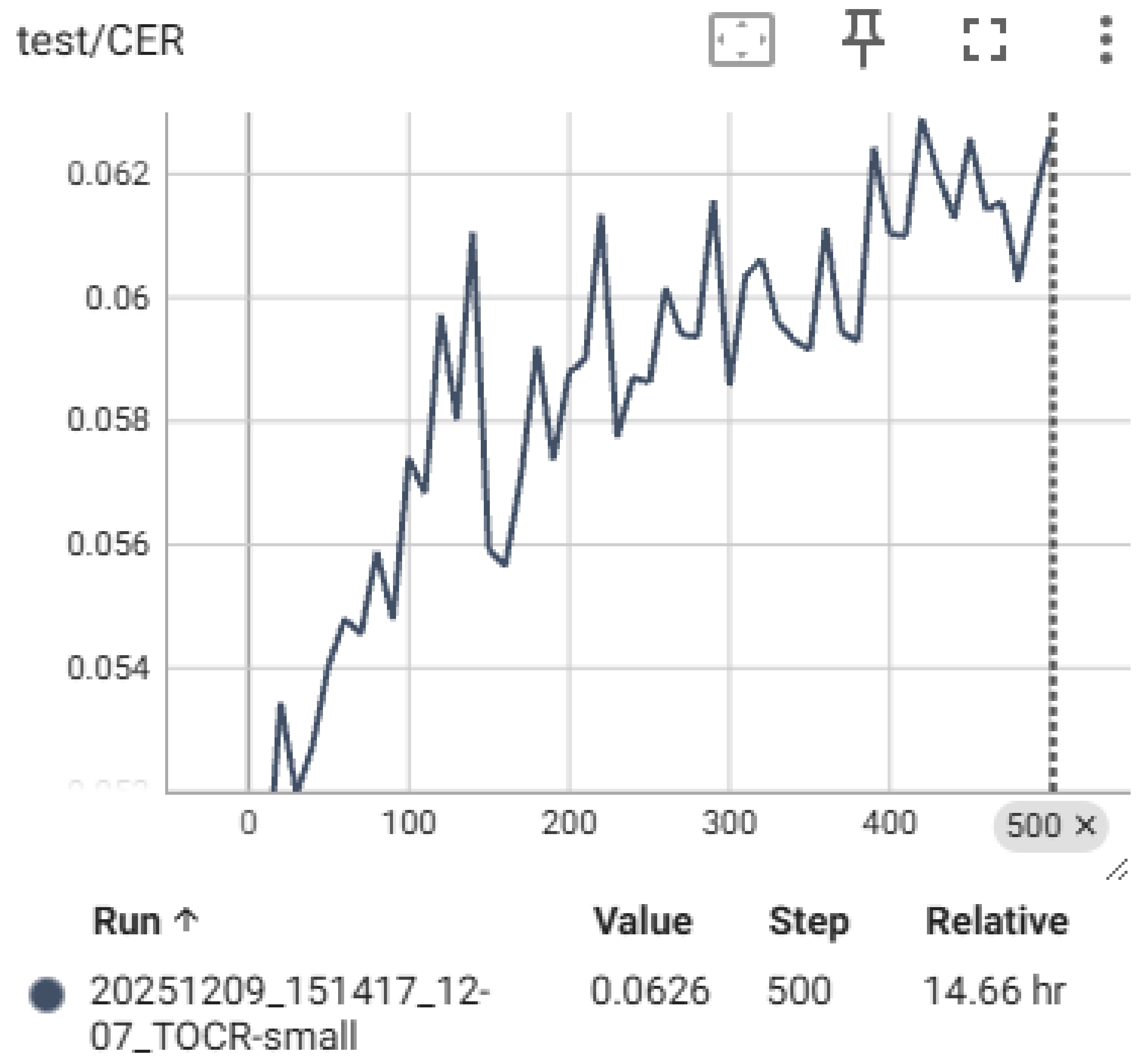
学習なしの評価→**CER:4.33**でいずれも精度悪化
lossは下がってるしもう少し学習ステップを増やしてみる→

Trocrファインチューニング編

ログ



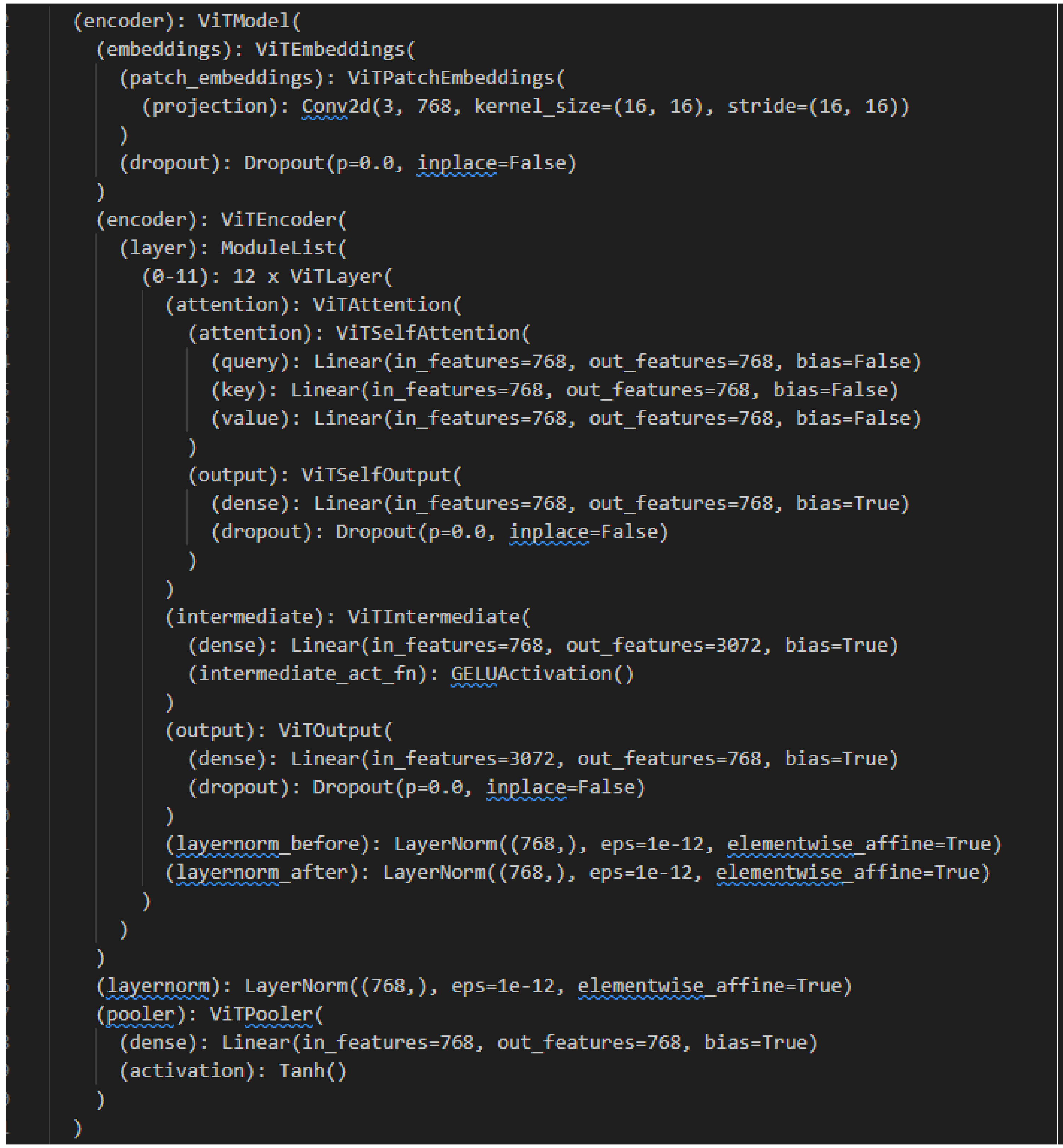
Run ↑	Value	Step	Relative
● 20251209_151417_12-07_TOCR-small	0	770,500	14.92 hr



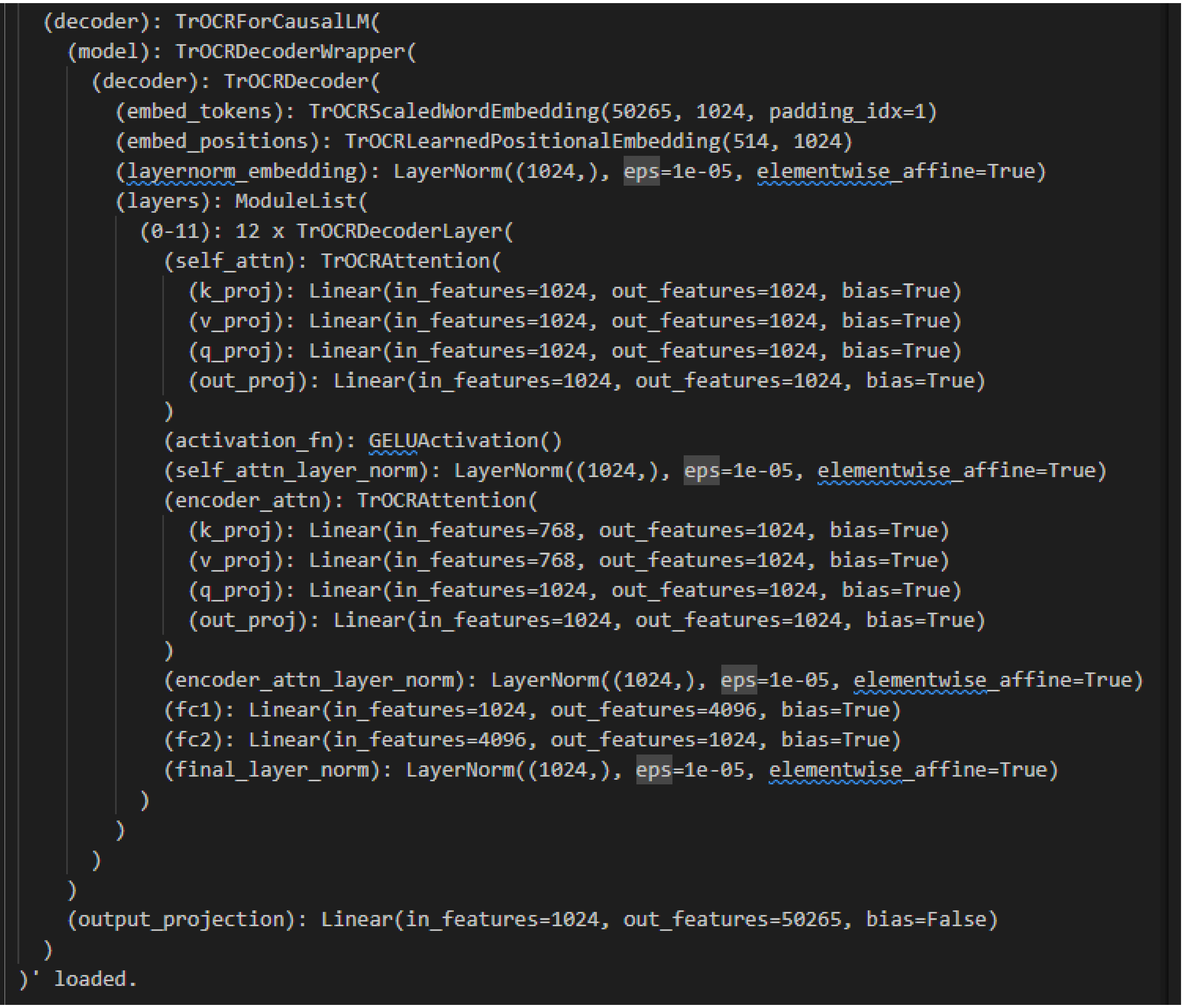
Run ↑	Value	Step	Relative
● 20251209_151417_12-07_TOCR-small	0.0626	500	14.66 hr

ステップごとの値とってしまった見にくいですよね...lol

モデル構造



```
  (encoder): ViTModel(
    (embeddings): ViTEmbeddings(
      (patch_embeddings): ViTPatchEmbeddings(
        (projection): Conv2d(3, 768, kernel_size=(16, 16), stride=(16, 16))
      )
      (dropout): Dropout(p=0.0, inplace=False)
    )
    (encoder): ViTEncoder(
      (layer): ModuleList(
        (0-11): 12 x ViTLayer(
          (attention): ViTAttention(
            (attention): ViTSelfAttention(
              (query): Linear(in_features=768, out_features=768, bias=False)
              (key): Linear(in_features=768, out_features=768, bias=False)
              (value): Linear(in_features=768, out_features=768, bias=False)
            )
            (output): ViTSelfOutput(
              (dense): Linear(in_features=768, out_features=768, bias=True)
              (dropout): Dropout(p=0.0, inplace=False)
            )
          )
          (intermediate): ViTIntermediate(
            (dense): Linear(in_features=768, out_features=3072, bias=True)
            (intermediate_act_fn): GELUActivation()
          )
          (output): ViTOutput(
            (dense): Linear(in_features=3072, out_features=768, bias=True)
            (dropout): Dropout(p=0.0, inplace=False)
          )
          (layernorm_before): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
          (layernorm_after): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
        )
      )
    )
    (layernorm): LayerNorm((768,), eps=1e-12, elementwise_affine=True)
    (pooler): ViTPooler(
      (dense): Linear(in_features=768, out_features=768, bias=True)
      (activation): Tanh()
    )
  )
```



```
  (decoder): TrOCRForCausalLM(
    (model): TrOCRDecoderWrapper(
      (decoder): TrOCRDecoder(
        (embed_tokens): TrOCRScaledWordEmbedding(50265, 1024, padding_idx=1)
        (embed_positions): TrOCRLearnedPositionalEmbedding(514, 1024)
        (layernorm_embedding): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
        (layers): ModuleList(
          (0-11): 12 x TrOCRDecoderLayer(
            (self_attn): TrOCRAttention(
              (k_proj): Linear(in_features=1024, out_features=1024, bias=True)
              (v_proj): Linear(in_features=1024, out_features=1024, bias=True)
              (q_proj): Linear(in_features=1024, out_features=1024, bias=True)
              (out_proj): Linear(in_features=1024, out_features=1024, bias=True)
            )
            (activation_fn): GELUActivation()
            (self_attn_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
            (encoder_attn): TrOCRAttention(
              (k_proj): Linear(in_features=768, out_features=1024, bias=True)
              (v_proj): Linear(in_features=768, out_features=1024, bias=True)
              (q_proj): Linear(in_features=1024, out_features=1024, bias=True)
              (out_proj): Linear(in_features=1024, out_features=1024, bias=True)
            )
            (encoder_attn_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
            (fc1): Linear(in_features=1024, out_features=4096, bias=True)
            (fc2): Linear(in_features=4096, out_features=1024, bias=True)
            (final_layer_norm): LayerNorm((1024,), eps=1e-05, elementwise_affine=True)
          )
        )
      )
    )
    (output_projection): Linear(in_features=1024, out_features=50265, bias=False)
  )
)' loaded.
```