

卒研に向けて

学会意見共有

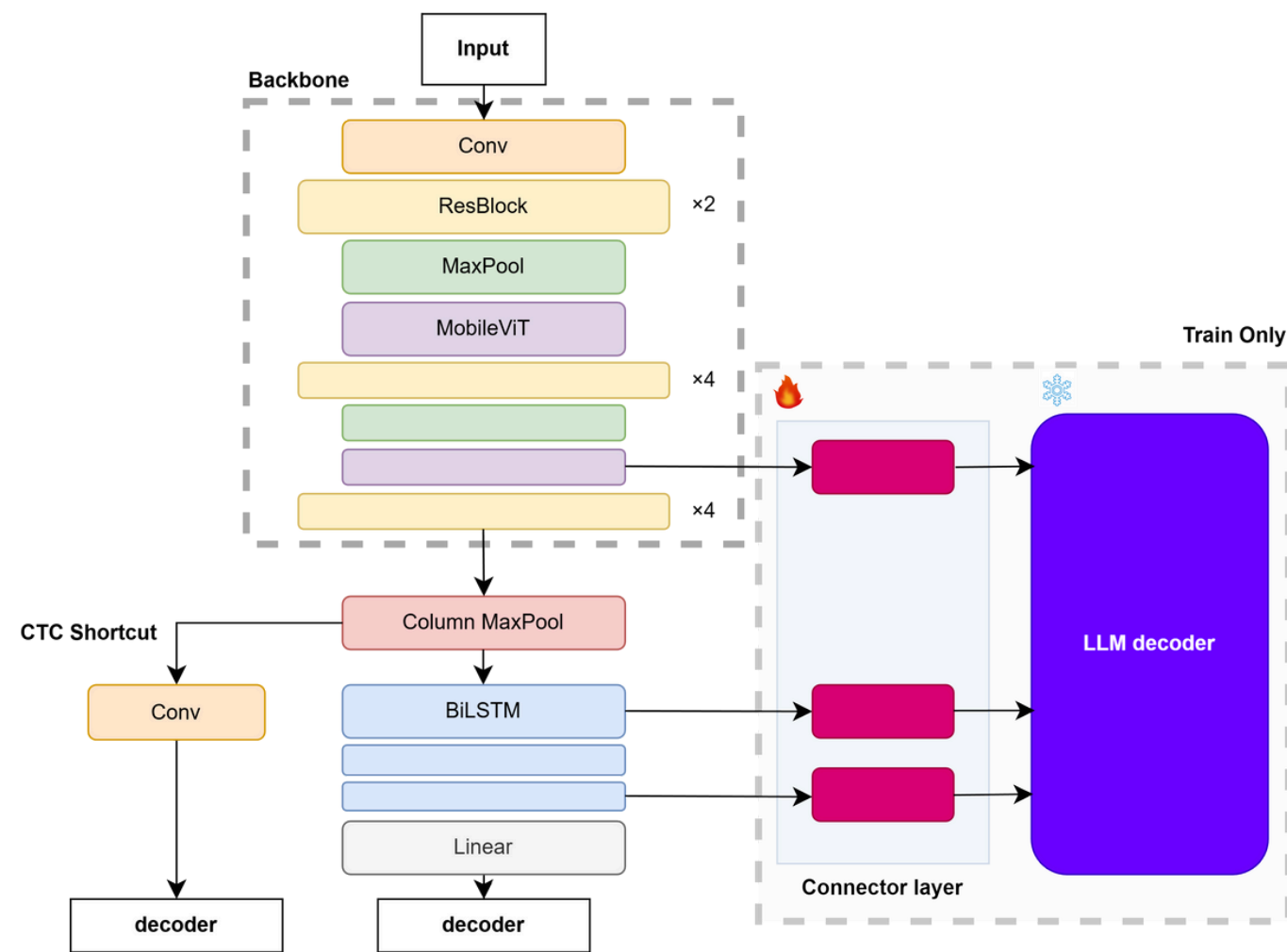
- データセットの筆者は一人か？
- 単語単位でも有用になるのはなぜか
- 精度がどれくらいになればよいとされているのか？

卒研どうしていくか

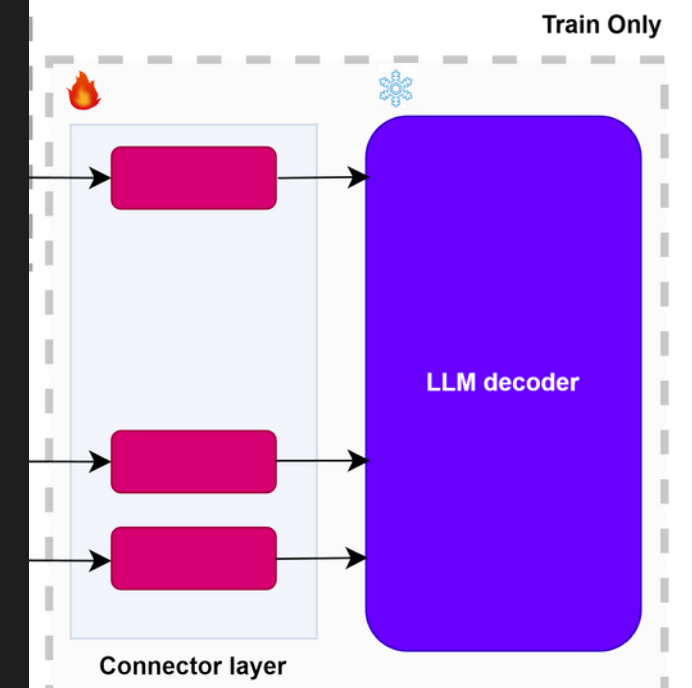
- タスクの変更
 - 言語的側面をいれるならより画像データが不明慮な状況
- なぜ精度が上昇しなかったのか
 - 言語特徴の付与させるためのエンコーダーとして不適切だったのでは
- どうするか
 - trocr をファインチューニングする
 - その学習時にLLM中間損失追加
 - でもtrocrのデコーダーgpt系使ってて言語的なものはとれてるかも??

卒研どうしていくか

具体的な変更内容



```
(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()  
)  
)
```



卒研どうしていくか

デコーダ内容

```
(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.
```

来週までにすること

1. ファインチューニングで論文記載の同水準の精度を出す

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

2. LLM中間損失の実装←前回の反省を生かす

a. GERUは使用しない

b. コネクタ層の学習時はtrocrを凍結

3. 2でうまく行った場合：さらなる工夫を思考

4. 1,2でうまく行かなかった場合学会の内容からさらに深堀

a. 例えば層を追加してみるetc