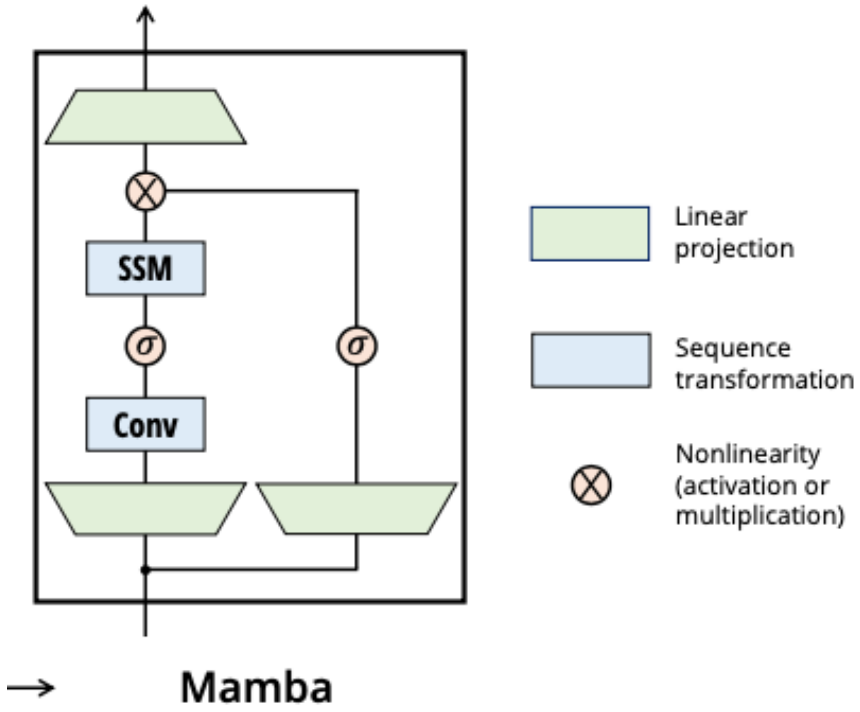
Implement detail
Train detail
Install Micromamba

Implement detail

- Q: Implementation of Discretization B issue 114 issue10
  - 代码中没有用$\bar{B}_t = (\Delta A)^{-1}(exp(\Delta A) - I) \cdot \Delta B$ 而是用了简化版的 ΔB (没有用原文中扯到的ZOH)

    > It makes the implementation slightly simpler without affecting empirical performance. One can think of it as a mix of ZOH for A and Euler discretization for B.

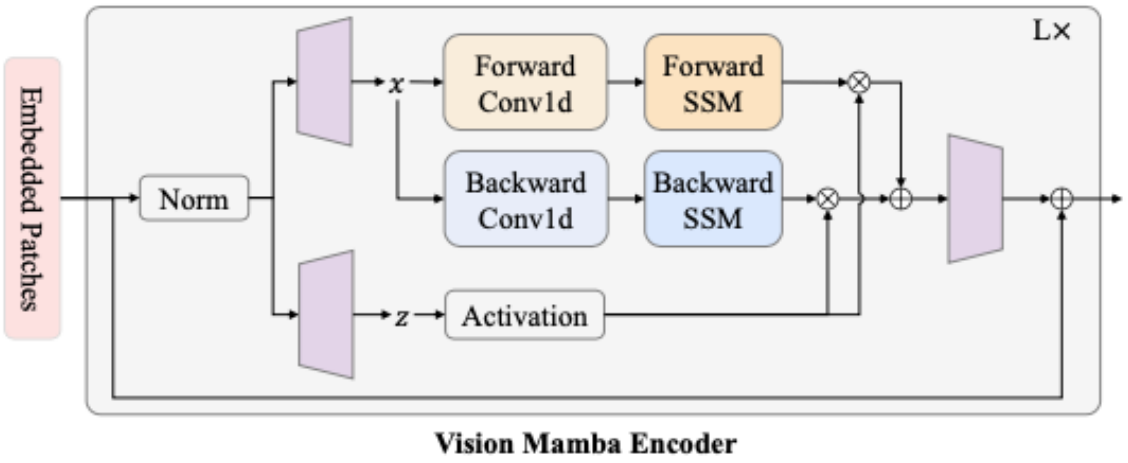
- Throughput related
  - cg=True matters issue90, to reduce CPU/IO
- Parameters



  - 一个mamba block的参数量 (D: model_size, N: dstate, e: expand_factor, K: conv1d_kernel_size)
    - $2eD^2 + eD(K+1) + eD(D/16 + 2N) + eD(D/16 + 1) + eDN + eD + eD^2$
    - 取e=2, 总参数量：$6.25D^2 + 5DN + 4D + 2D(K+1)$
    - 取N=16, K=4, 总参数量: $6.25D^2 + 94D$
    - In detail
      - Conv block:

        $$\#params = (kernel\_size \times |\frac{in\_channel}{groups}| + 1) \times |out\_channels|$$

- 双向mamba



Vision Mamba Encoder

  - $6D^2 + 2(2D(K+1) + 0.25D^2 + 5DN + 4D)$
  - 取N=16, K=4, 总参数量: $6.5D^2 + 188D$
- For reference: each transformer layer: $12D^2$

Train detail

1. language modeling - scaling law (补充信息来自issue144)

1. Table 12: (**Scaling Law Model Sizes.**) Our model sizes and hyperparameters for scaling experiments. (Model dimension and number of heads applies only to Transformer models.)

Params	n_layers	d_model	n_heads / d_head	Training steps	Learning Rate	Batch Size	Tokens
125M	12	768	12 / 64	4800	6e-4	0.5M tokens	2.5B
350M	24	1024	16 / 64	13500	3e-4	0.5M tokens	7B
760M	24	1536	16 / 96	29000	2.5e-4	0.5M tokens	15B
1.3B	24	2048	32 / 64	50000	2e-4	0.5M tokens	26B

   注意这里的lr是base，实际会x5

   - 2.8B model lr = 3xGPT3(1.6x1e-4) = 4.8x1e-4
2. 原文Train recipes中写道

> gradient clip = 1.0; weight decay = 0.1; no dropout; linear lr warmup + cosine decay (decay to 1e-5, peak value set $5\times$ GPT3 value)
>
> - 参考GPT-3的scaling设定
>
> | Model Name | n_{params} | n_{layers} | d_{model} | n_{heads} | d_{head} | Batch Size | Learning Rate |
> |---|---|---|---|---|---|---|---|
> | GPT-3 Small | 125M | 12 | 768 | 12 | 64 | 0.5M | 6.0×10^{-4} |
> | GPT-3 Medium | 350M | 24 | 1024 | 16 | 64 | 0.5M | 3.0×10^{-4} |
> | GPT-3 Large | 760M | 24 | 1536 | 16 | 96 | 0.5M | 2.5×10^{-4} |
> | GPT-3 XL | 1.3B | 24 | 2048 | 24 | 128 | 1M | 2.0×10^{-4} |
> | GPT-3 2.7B | 2.7B | 32 | 2560 | 32 | 80 | 1M | 1.6×10^{-4} |
> | GPT-3 6.7B | 6.7B | 32 | 4096 | 32 | 128 | 2M | 1.2×10^{-4} |
> | GPT-3 13B | 13.0B | 40 | 5140 | 40 | 128 | 2M | 1.0×10^{-4} |
> | GPT-3 175B or "GPT-3" | 175.0B | 96 | 12288 | 96 | 128 | 3.2M | 0.6×10^{-4} |
>
> **Table 2.1:** Sizes, architectures, and learning hyper-parameters (batch size in tokens and learning rate) of the models which we trained. All models were trained for a total of 300 billion tokens.
>
> no bias term: RMSNorm (instead of LayerNorm) AdamW $\beta = (0.9, 0.95)$

Install Micromamba

```
# py39 torch211+cu118: /cto_labas/AIDD/mamba/vllm.yaml

# 1. create new env [Name: prot]
1.9.0 prot python==3.9

# 2. install pytorch related
## pytorch version: 2.1.1+cu118
pip3 install torch==2.1.1+cu118 torchvision==0.16.1+cu118 -f
https://download.pytorch.org/whl/torch_stable.html

## pytorch verison: 1.13.1+cu116
pip3 install torch==1.13.1+cu116 torchvision==0.14.1+cu116 -f
https://download.pytorch.org/whl/torch_stable.html
```