band_attention

A simple implementation of band attention with CUDA acceleration for faster Diffusion Transformers in sequential generation task.

Motivation

Installation

This package can be installed on Linux with

```
git clone https://github.com/luoshuqing2001/band_attention.git
cd band_attention
python pytorch/setup.py install
```

Usage

```
import torch
import band_attention

device = "cuda:0"
Q = torch.randn(bs, nh, nt, channel, dtype=torch.float32,
device=device) ## Query Tensor
K = torch.randn(bs, nh, nt, channel, dtype=torch.float32,
device=device) ## Key Tensor
V = torch.randn(bs, nh, nt, channel, dtype=torch.float32,
device=device) ## Value Tensor
attn = torch.zeros(bs, nh, nt, nt, dtype=torch.float32, device=device)
## Attention Tensor
X = torch.zeros(bs, nh, nt, channel, dtype=torch.float32,
device=device) ## Result Tensor

band_attention.torch_launch_band_attention(X.reshape(-1), \
                attn.reshape(-1), Q.reshape(-1), K.reshape(-1),
V.reshape(-1), \
                window, bs, nh, nt, channel)

X = X.reshape(bs, nh, nt, channel)
```

Implementation Details

We implement band attention using CUDA acceleration. Details can be seen at Details

Results

This package can achieve 2+ times speed up compared with self attention and 100+ times speed up with masked self attention in average. With a relatively small value of `Window`, it can save 10+ times FLOPs during inference.