Overview

Model

- input images and labels: dimensions (sequence length, batch size, embedding dim)
- train mode:
  - add start token (all zeros) to beginning of labels, remove last label token (sequence length remains the same)
  - positional encoding of images and labels (sin/cos method)
  - normalization of images and labels (each token normalized to L2 norm of 1)
  - creation of additive mask (dimensions (sequence length, sequence length) used for decoder self-attention: $\begin{bmatrix} 0 & -\infty & \dots & -\infty \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & -\infty \end{bmatrix}$
  - feeding images and labels into pytorch transformer (8 heads for multihead attention, 6 encoder layers, 6 decoder layers)
  - if cross-entropy loss is used: 2D convolution layer applied to transformer output (mapping outputs from dimension (batch size, 1, sequence length, embedding dim) to logits with dimension (batch size, 4, sequence length, embedding dim))
- inference mode:
  - initializing target to sequence only containing the start token
  - positional encoding of images and target (sin/cos method)
  - normalization of images (each token normalized to L2 norm of 1)
  - produce outputs one token at a time:
    - normalization of target (each token normalized to L2 norm of 1)
    - feeding images and target into pytorch transformer (8 heads for multihead attention, 6 encoder layers, 6 decoder layers)
    - if cross-entropy loss is used: 2D convolution layer applied to transformer output (mapping outputs from dimension (batch size, 1, sequence length, embedding dim) to logits with dimension (batch size, 4, sequence length, embedding dim))
    - positional encoding of latest output token
    - append latest output token to target