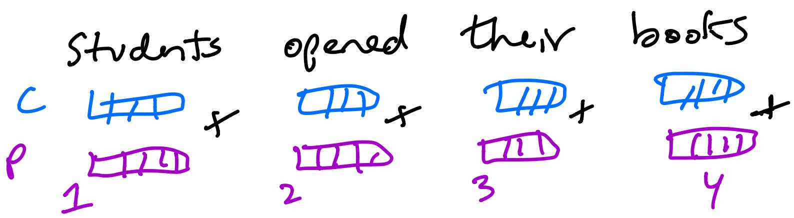
Position embeddings in Transformers:

- without some explicit injection of position info, self-attention doesn't have any notion of order



{ fixed → orig Transformer paper
{ learned → BERT

↳ additive pos. embs
absolute pos. embs

absolute vs. relative pos embs
↳ represent every pair of tokens in the input

$$q_{students} = W_q \cdot (c_{students} + p_1)$$

---

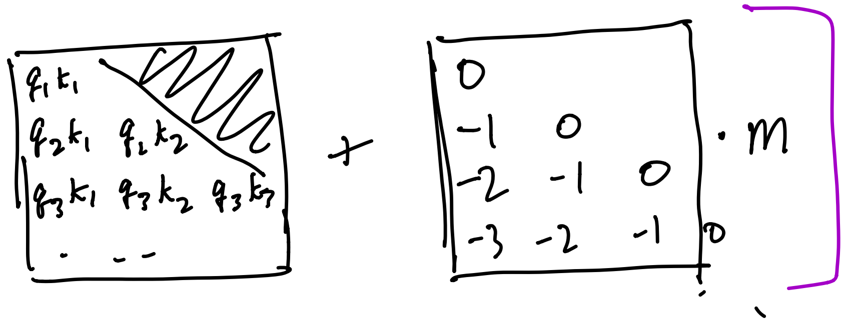
relative position embs:

→ generally cannot be added directly to input embs (RoPE is an exception)

→ instead directly modify the attn matrix

ALiBi:

$$q_{students} = f(W_q \cdot c_{students})$$
$$k_{books} = f(W_k \cdot c_{books})$$



fixed, m is a hyperparam
varies by head
$m = \frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \ldots$

⇒ intuitively: words that are closer together have a higher dot product

⇒ ALiBi enables extrapolation beyond the training seq length

⇒ position info is only affecting q, k, but not v

Rotary position embs (RoPE)

- enables relative pos. embs without modifying the attn matrix like ALiBi
- instead of adding pos. emb, we actually rotate the q, k vectors via matrix/vector product w/ a rotation matrix

- goal: dot product of rotated q, k ($q^T k$) should be a function of relative position only, not abs. pos

ex: c_1 c_2 c_3 c_4

students opened their books

we want to compute $\underline{q_4} \cdot k_1$

RoPE: find f_q, f_k, g such that

$$f_q(c_{books}, 4) = q_4$$
$$f_k(c_{students}, 1) = k_1$$
$$q_4 \cdot k_1 = g(c_{books}, c_{students}, 3)$$

↳ 4-1

⇒ this can be accomplished by rotating $W_q c_i$ and $W_k c_i$ by diff. angles

$$f_q(c_{books}, 4) = R_{\theta,4} \cdot W_q c_{books}$$
$$f_k(c_{students}, 1) = R_{\theta,1} \cdot W_k c_{students}$$

$$g = q^T k$$

$$R_{\theta, t} = \begin{bmatrix} \cos t\theta & -\sin t\theta \\ \sin t\theta & \cos t\theta \end{bmatrix}$$

where θ is hyperparameter