RoPE and WRoPE

- Rotational Positional Encoding (RoPE) owns *one arc direction* along the hypersphere
- We can thus rotate our vector memory $h(n)$ by $\Delta$ radians each time step to “age” it:

  $$h_{\alpha}(n) = e^{j\Delta n}, \quad \text{with} \quad \Delta = \frac{2\pi}{L}$$

  when our maximum sequence length (before reset) is $L$
- **Idea:** “Warped RoPE” (WRoPE) for *arbitrarily long sequences* (processed in reverse):

  $$\Delta_n = \frac{2\pi n}{n + L}, \quad n = 0, 1, 2, \ldots$$

  (inspired by the *bilinear transform* used in digital filter design)
- A *blend* of uniform and warped rotations can be used:

  $$\Delta_n = \begin{cases} \frac{\pi n}{L}, & n = 0, 1, 2, \ldots, L - 1 \\ \pi + \frac{\pi n}{n + 1}, & n = L, L + 1, L + 2, \ldots \end{cases}$$

  where $L$ is now the *typical* sequence length (giving it more “space” in recall)
WRoPE Memory

- WRoPE sequences are naturally reversed because we can only change all stored angles by the same delta:

\[ h_a(n) = e^{j\Delta n} h(n), \quad n = 0, 1, 2, \ldots \]

- This makes inference non-autoregressive (more expensive)

- One improvement is to store past hidden states so that positional encodings can be updated arbitrarily when accessed:

\[ h_a(n, m) = e^{j\Delta n-m} h(m), \quad m = n - L, \ldots, n - 1, n \]

\((m^{th} \text{ hidden state vector needed for inference at time } n)\)

- This is the same amount of storage needed for the Truncated Infinite Impulse Response (TIIR) technique which provides a recursively computed sliding-window of memory

- In the TIIR case (fixed length \(L\)), might as well use normal RoPE

- WRoPE maybe competitive for encoding “journalistic style” into a vector
Truncated Infinite Impulse Response (TIIR) RNNs (TRNN)

A *sliding rectangular window* can be obtained as an integrator minus a *delayed* integrator:

\[ [1, 1, \ldots, 1] \leftrightarrow \sum_{n=0}^{N-1} z^{-n} = \frac{1 - z^{-N}}{1 - z^{-1}} = \frac{1}{1 - z^{-1}} - z^{-N} \frac{1}{1 - z^{-1}} \]

- Thus, two identical RNNs can be *differenced* to provide a non-fading, linearly RoPEd memory of any length \( L \)
- A *real* memory of length \( L \) is needed for the *hidden state update*:
  \[ dh(n) = h(n + 1) - h(n) = B_n x(n) \]
- Hidden state update becomes
  \[ h(n + 1) = h(n) + dh_n = h(n) + B_n x(n) - B_{n-L} x(n - L) \]
TRNN with Sliding-Window Memory and Linear RoPE

* Optional Attention Sum