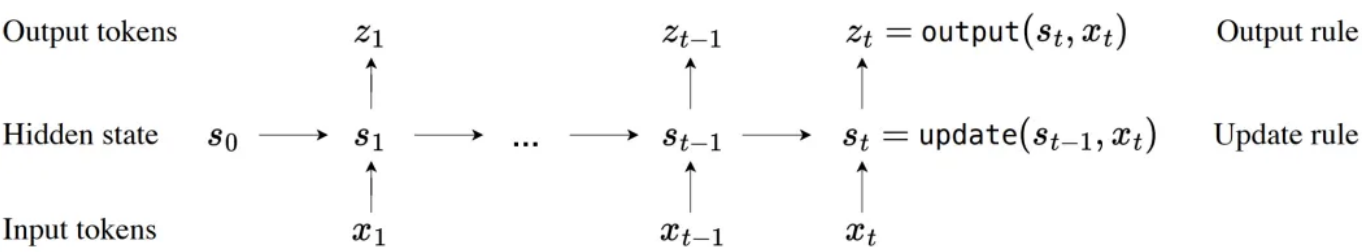


# TTT - Learning to (Learn at Test Time)



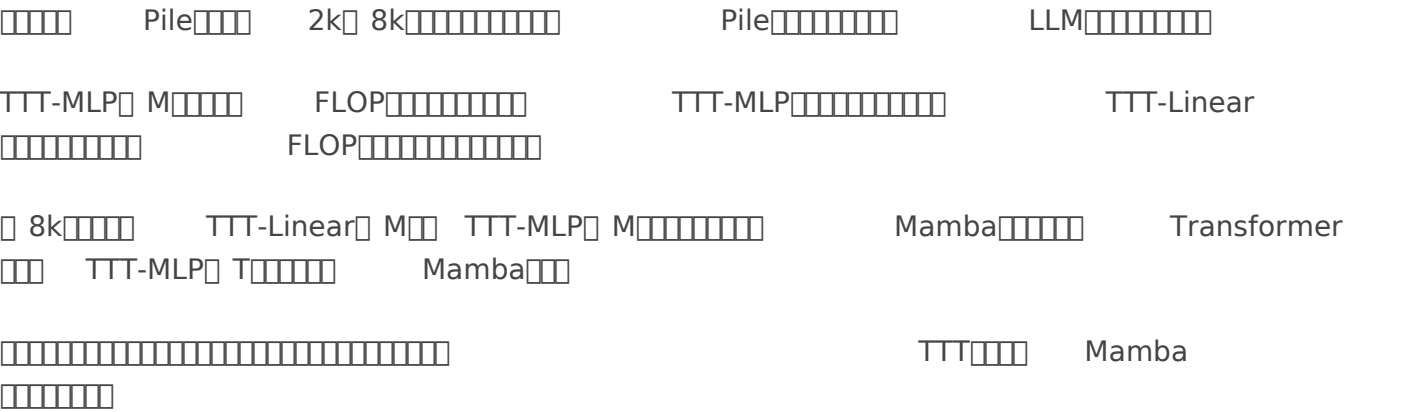
	Initial state	Update rule	Output rule	Cost
Naive RNN	$s_0 = \text{vector}()$	$s_t = \sigma(\theta_{ss}s_{t-1} + \theta_{sx}x_t)$	$z_t = \theta_{zs}s_t + \theta_{zx}x_t$	$O(1)$
Self-attention	$s_0 = \text{list}()$	$s_t = s_{t-1}.\text{append}(k_t, v_t)$	$z_t = V_t \text{softmax}(K_t^T q_t)$	$O(t)$
Naive TTT	$W_0 = f.\text{params}()$	$W_t = W_{t-1} - \eta \nabla \ell(W_{t-1}; x_t)$	$z_t = f(x_t; W_t)$	$O(1)$

Figure 4. **Top:** A generic sequence modeling layer expressed as a hidden state that transitions according to an update rule. All sequence modeling layers can be viewed as different instantiations of three components in this figure: the initial state, update rule and output rule. **Bottom:** Examples of sequence modeling layers and their instantiations of the three components. The naive TTT layer was shown in Figure 1. Self-attention has a hidden state growing with context, therefore growing cost per token. Both the naive RNN and TTT layer compress the growing context into a hidden state of fixed size, therefore their cost per token stays constant.





??



Revision #1  
Created 11 January 2025 09:44:05 by Colin  
Updated 11 January 2025 09:44:09 by Colin