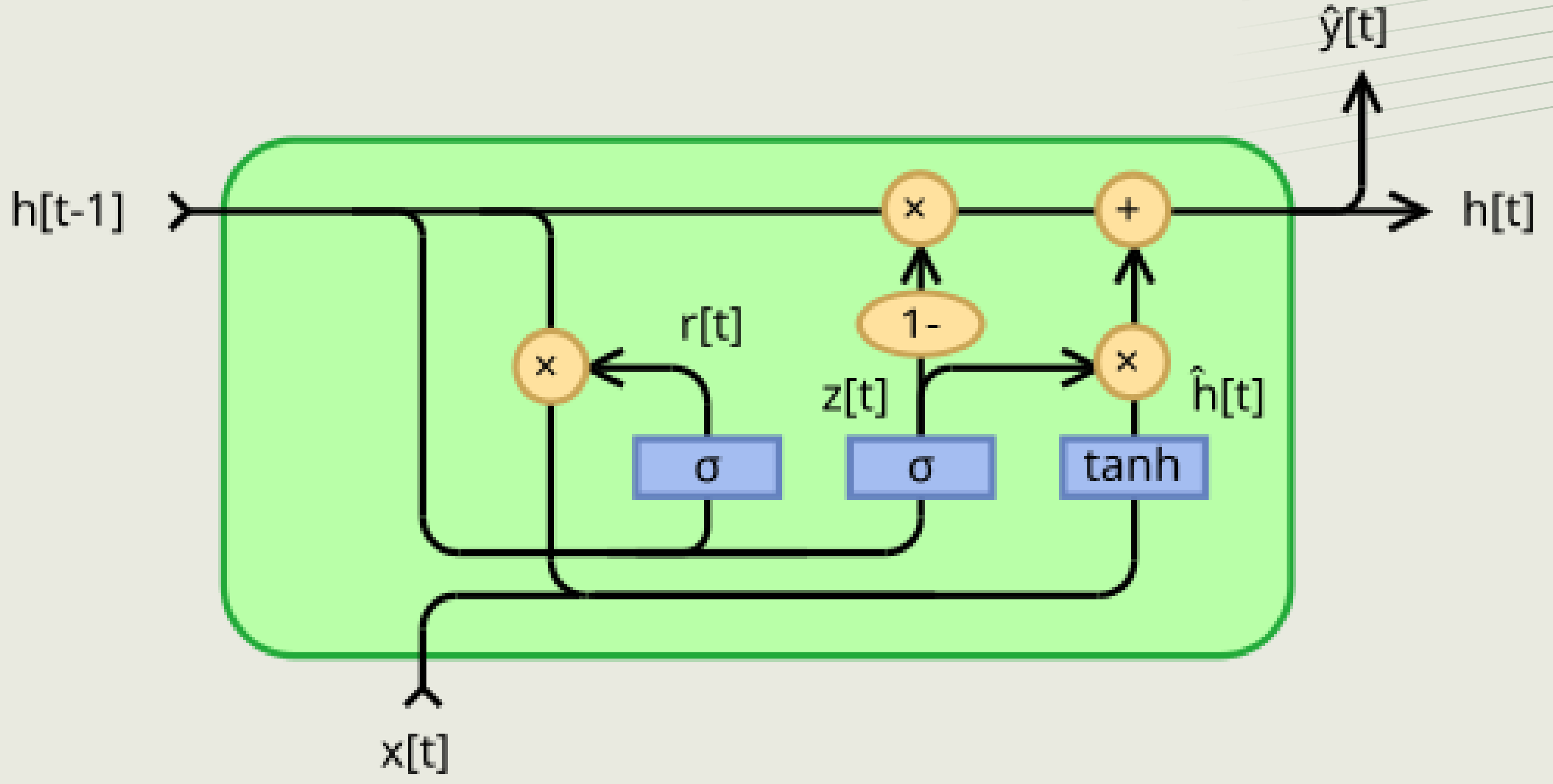
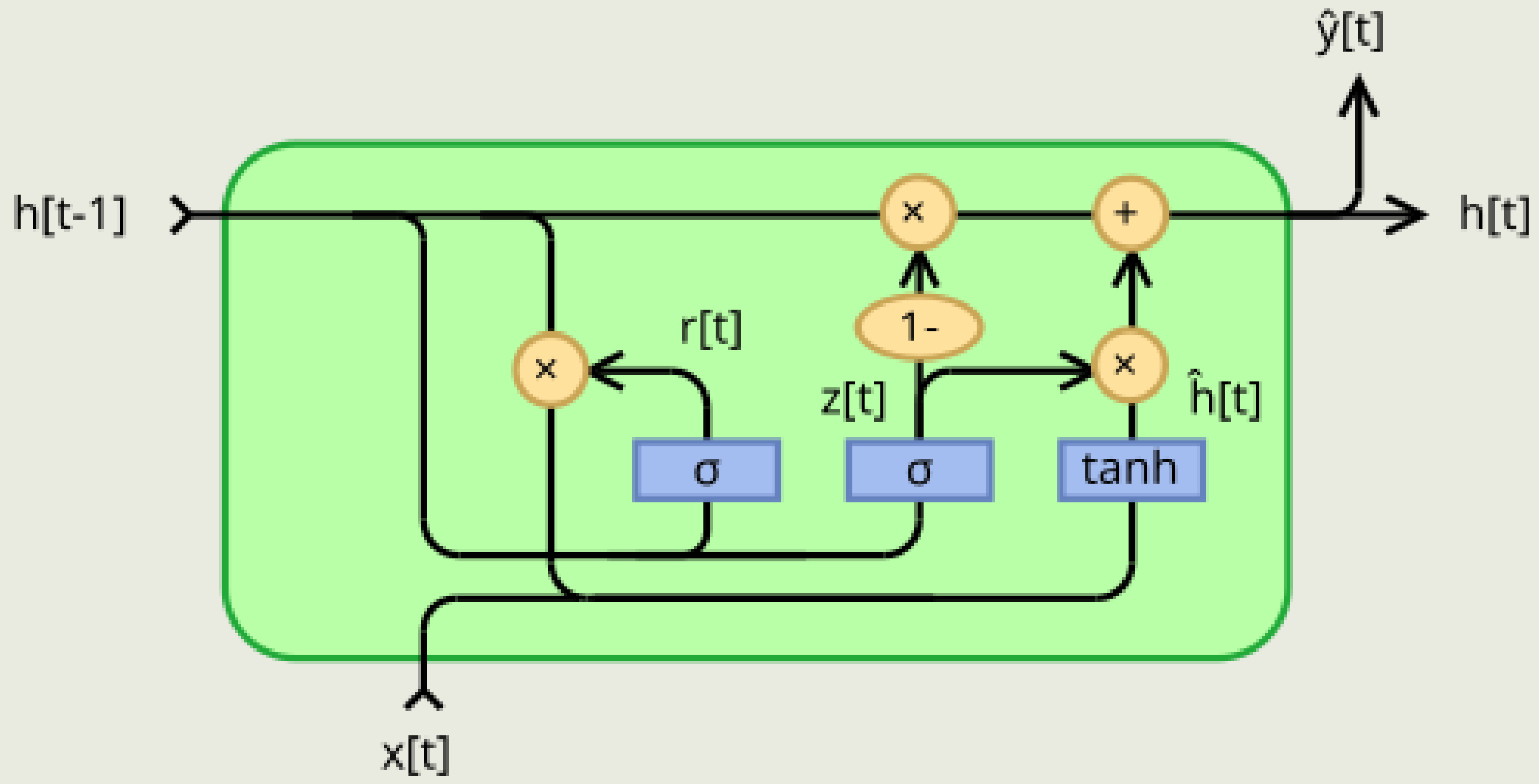
SEQUENCE ANALYSIS WITH MIN RNN

BY Mohan Sharma

GRU NETWORKS



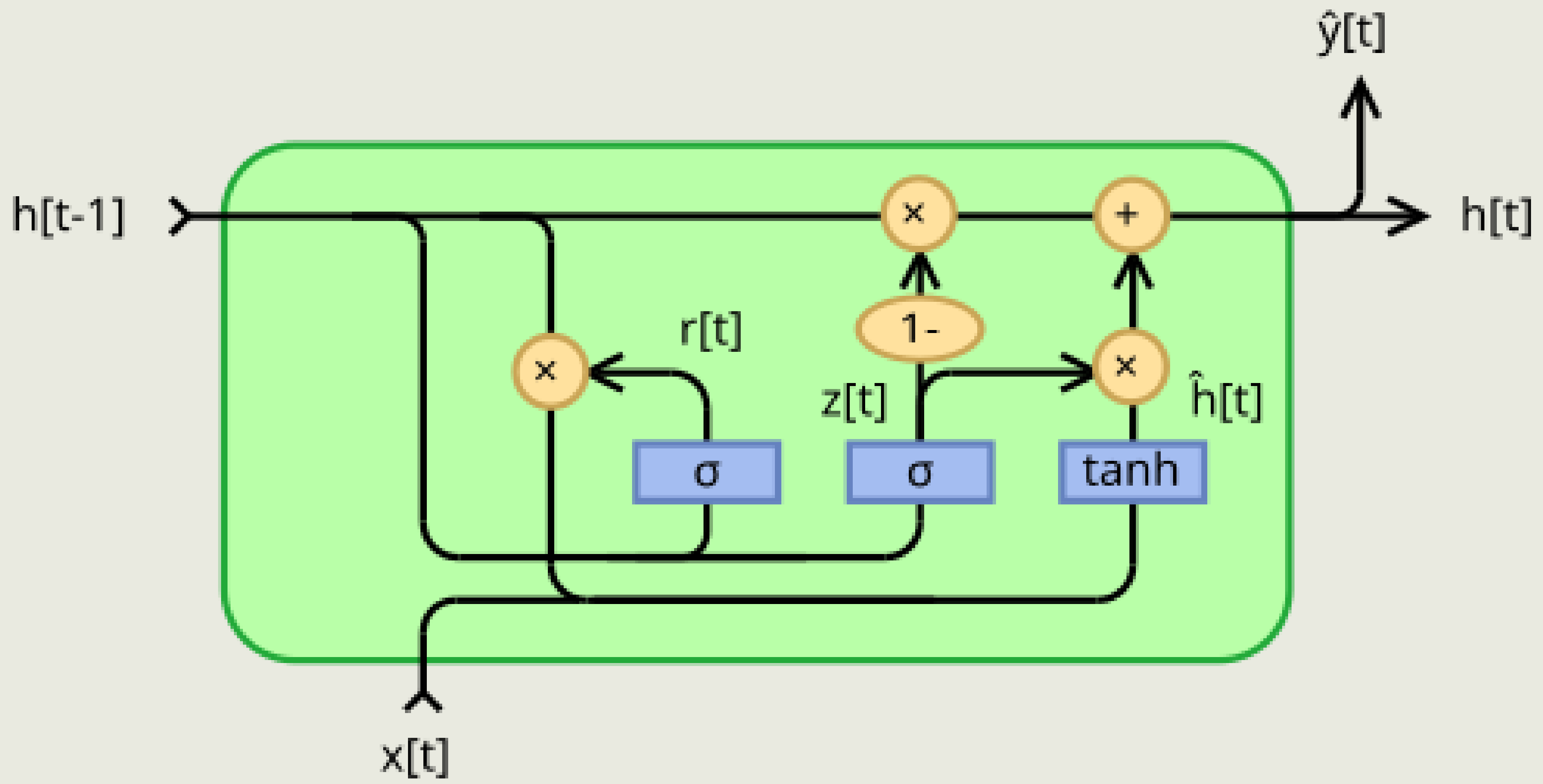
STEP-BY-STEP GRU WALK THROUGH



r_t = sigmoid(W_r * [h_t-1, x_t])
z_t = sigmoid(W_z * [h_t-1, x_t])Gate

1. The first step in our LSTM is to decide what informatioThe reset gate r and update gate z are computed using the current input x and the previous hidden state h_t-1

n we're going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer."



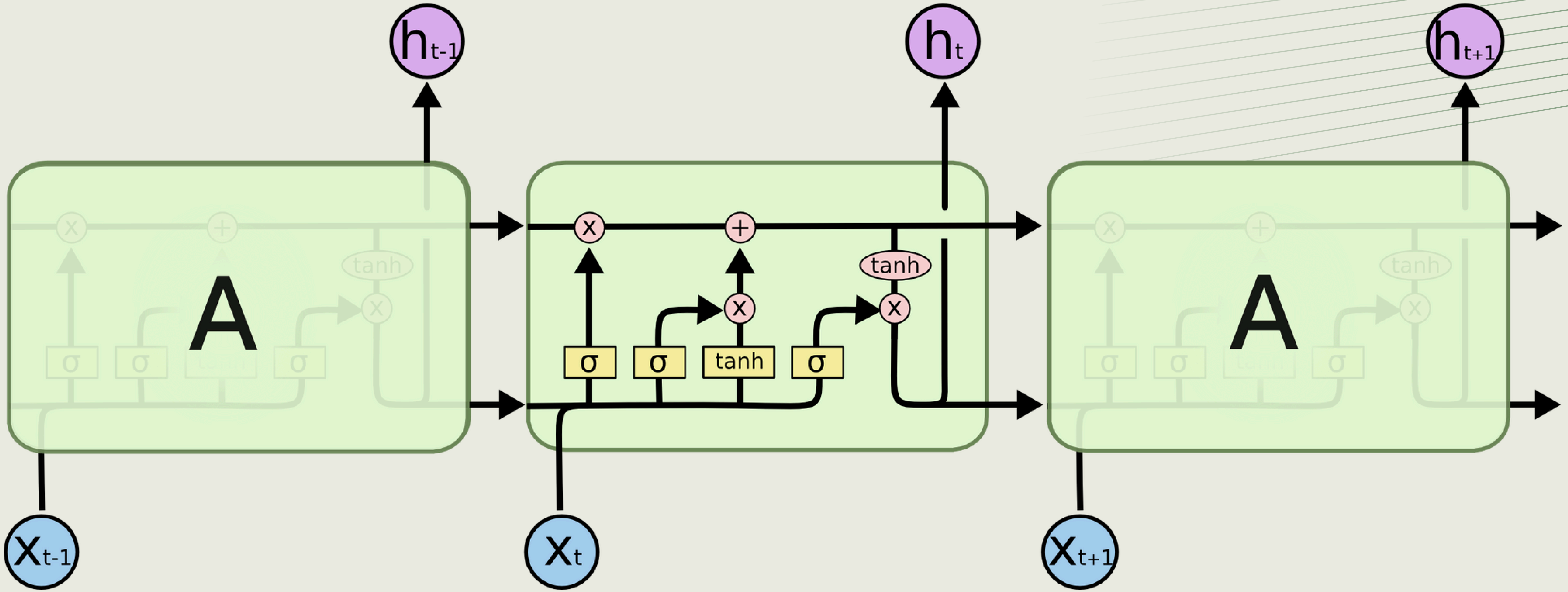
2) The candidate activation vector h_t~ is computed using the current input x and a modified version of the previous hidden state that is "reset" by the reset gate

$$h\_t\sim = \tanh(W\_h * [r\_t * h\_t\text{-}1, x\_t])$$

3) The new hidden state h_t is computed by combining the candidate activation vector with the previous hidden state, weighted by the update gate

$$h\_t = (1 - z\_t) * h\_t\text{-}1 + z\_t * h\_t\sim$$

LSTM NETWORKS



LIMITATIONS

- Computational Cost and time is high
- Parallelization is not possible
- Performance on Long Sequences is not so good
- Memory Inefficiency

MIN GRU

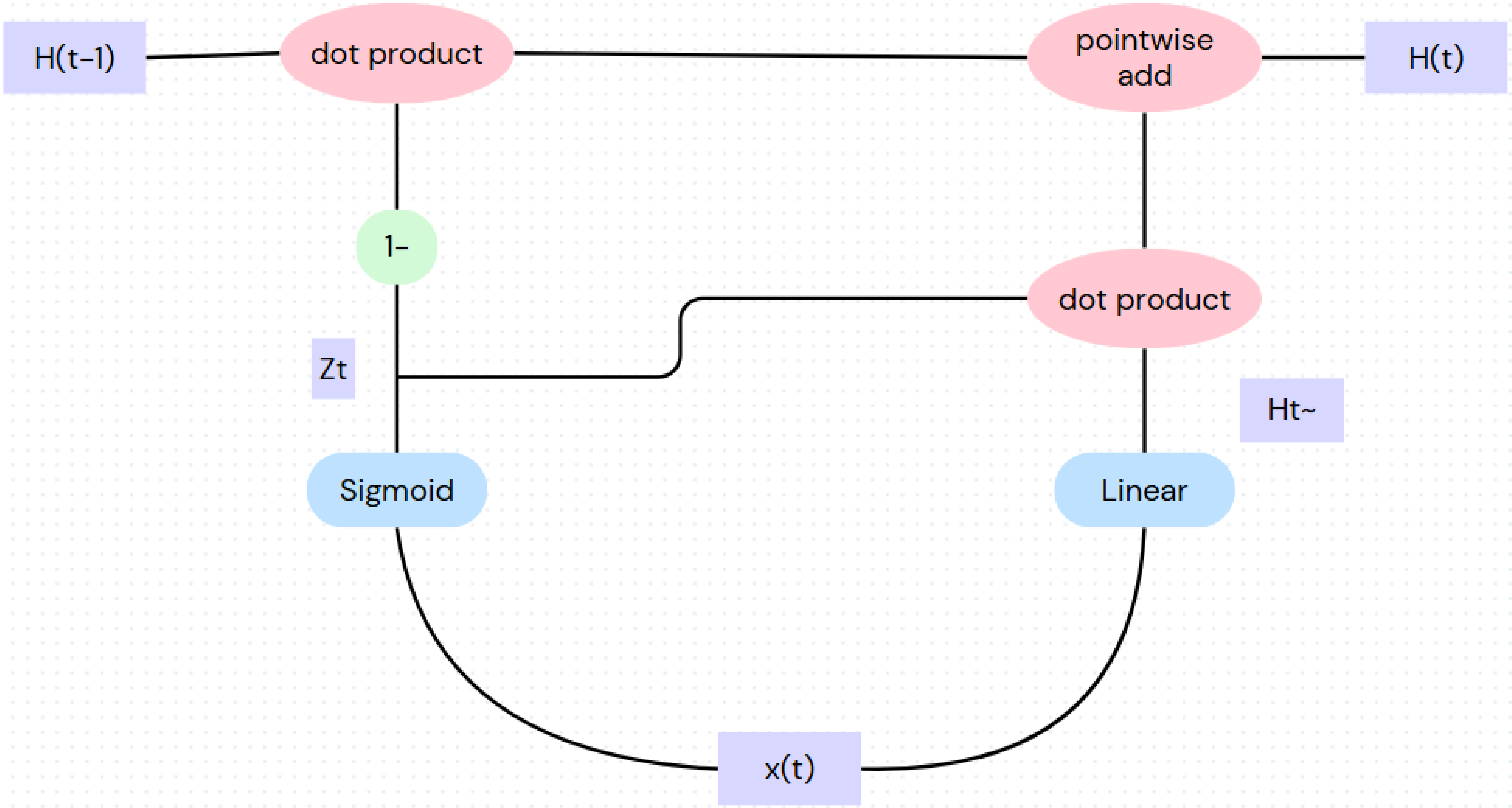
GRU

$$
\begin{aligned}
\boldsymbol{h}_t &= (\boldsymbol{1} - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t \\
\boldsymbol{z}_t &= \sigma(\text{Linear}_{d_h}([\boldsymbol{x}_t, \boldsymbol{h}_{t-1}])) \\
\boldsymbol{r}_t &= \sigma(\text{Linear}_{d_h}([\boldsymbol{x}_t, \boldsymbol{h}_{t-1}])) \\
\tilde{\boldsymbol{h}}_t &= \tanh(\text{Linear}_{d_h}([\boldsymbol{x}_t, \boldsymbol{r}_t \odot \boldsymbol{h}_{t-1}]))
\end{aligned}
$$

$\Rightarrow$

minGRU

$$
\begin{aligned}
\boldsymbol{h}_t &= (\boldsymbol{1} - \boldsymbol{z}_t) \odot \boldsymbol{h}_{t-1} + \boldsymbol{z}_t \odot \tilde{\boldsymbol{h}}_t \\
\boldsymbol{z}_t &= \sigma(\text{Linear}_{d_h}(\boldsymbol{x}_t)) \\
\tilde{\boldsymbol{h}}_t &= \text{Linear}_{d_h}(\boldsymbol{x}_t)
\end{aligned}
$$


H(t-1)
dot product
pointwise add
H(t)
1-
dot product
Zt
Ht~
Sigmoid
Linear
x(t)

MIN LSTM

To remove the previous dependencies by decomposing to Model to bare Bones

LSTM

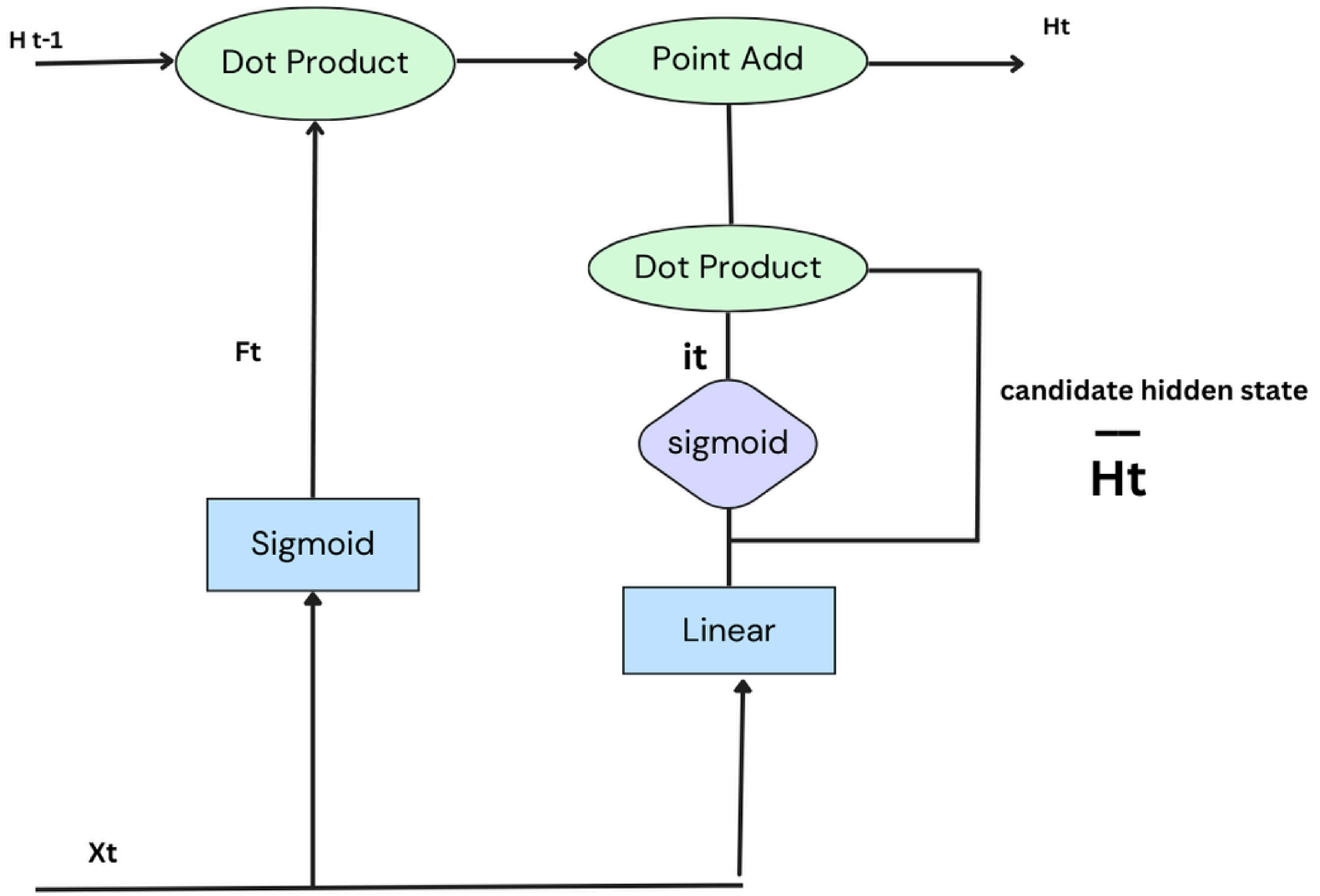
$$
\begin{aligned}
\boldsymbol{h}_t &= \boldsymbol{o}_t \odot \tanh(\boldsymbol{c}_t) \\
\boldsymbol{o}_t &= \sigma(\mathrm{Linear}_{d_h}([\boldsymbol{x}_t, \boldsymbol{h}_{t-1}])) \\
\boldsymbol{c}_t &= \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t \\
\boldsymbol{f}_t &= \sigma(\mathrm{Linear}_{d_h}([\boldsymbol{x}_t, \boldsymbol{h}_{t-1}])) \\
\boldsymbol{i}_t &= \sigma(\mathrm{Linear}_{d_h}([\boldsymbol{x}_t, \boldsymbol{h}_{t-1}])) \\
\tilde{\boldsymbol{c}}_t &= \tanh(\mathrm{Linear}_{d_h}([\boldsymbol{x}_t, \boldsymbol{h}_{t-1}]))
\end{aligned}
$$

$\Rightarrow$

minLSTM

$$
\begin{aligned}
\boldsymbol{h}_t &= \boldsymbol{f}_t \odot \boldsymbol{h}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{h}}_t \\
\boldsymbol{f}_t &= \sigma(\mathrm{Linear}_{d_h}(\boldsymbol{x}_t)) \\
\boldsymbol{i}_t &= \sigma(\mathrm{Linear}_{d_h}(\boldsymbol{x}_t)) \\
\tilde{\boldsymbol{h}}_t &= \mathrm{Linear}_{d_h}(\boldsymbol{x}_t)
\end{aligned}
$$

- Remove cell state
- Remove previous dependencies on previous hidden and cell state



HOW TO COMPUTE THIS IN PARALLEL??

PARALLEL SCAN ALGORITHM

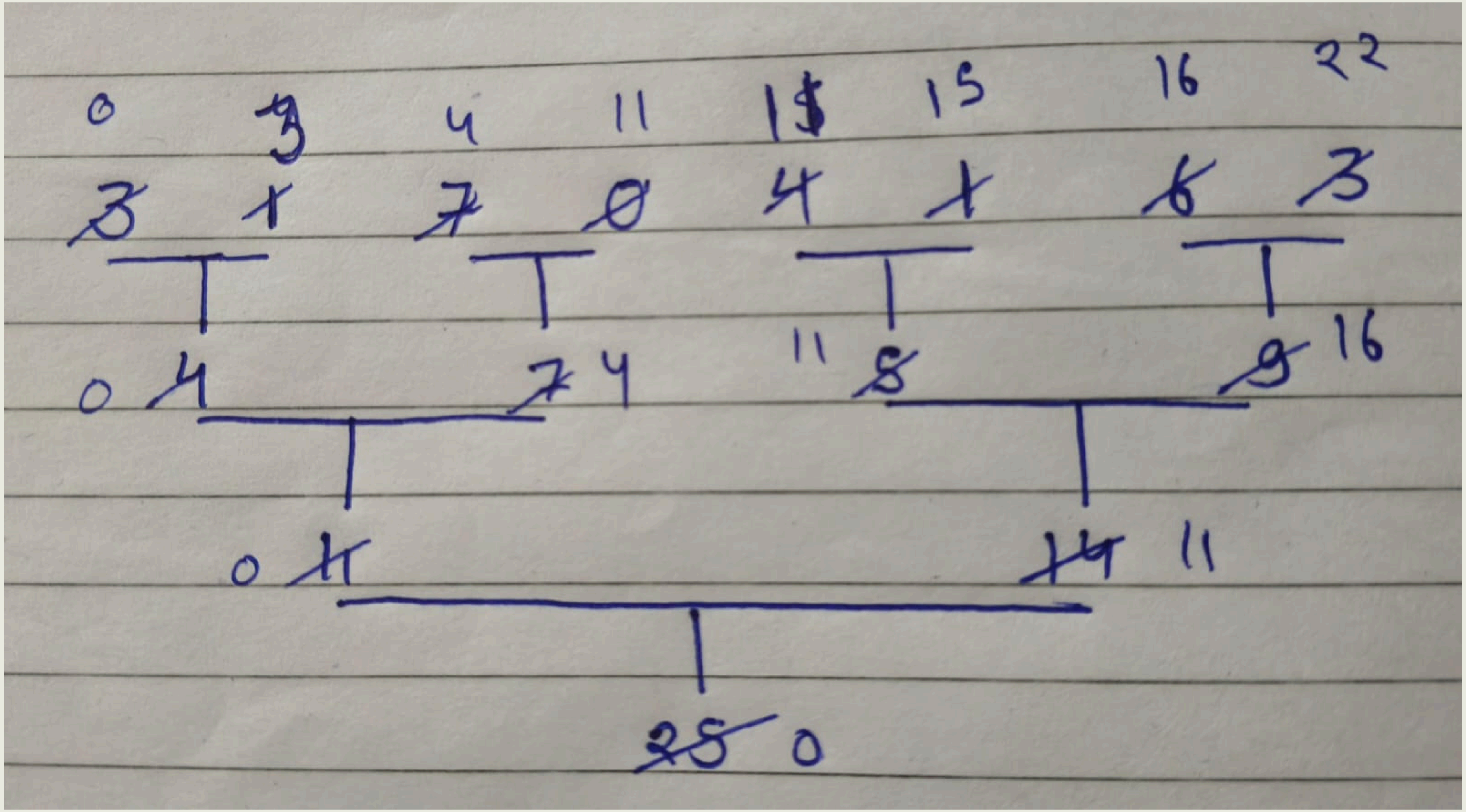
1. UP-SWEEP PHASE (REDUCE)

BUILD A BINARY TREE FROM THE ARRAY BOTTOM-UP BY SUMMING PAIRS OF ELEMENTS:

- AT EACH LEVEL, EVERY PROCESSOR WORKS ON 2 ELEMENTS AND STORES THE SUM IN THE PARENT NODE.
- THE LAST ELEMENT WILL HAVE THE TOTAL SUM OF THE ARRAY.

2. DOWN-SWEEP PHASE

- SET THE ROOT NODE TO 0.
- TRAVERSE THE TREE BACK DOWN:
    - SWAP VALUES AND COMPUTE PREFIX SUMS USING VALUES FROM THE UP-SWEEP PHASE.



INPUT -3 1 7 0 4 1 6 3

INPUT - 0 3 4 11 11 12 16 22
(exclusive prefix sum)

How Is the parallel sum algo is applied to min GRU and LSTM?

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t$$

As

$$h_t = a_t \cdot h_{t-1} + b_t$$

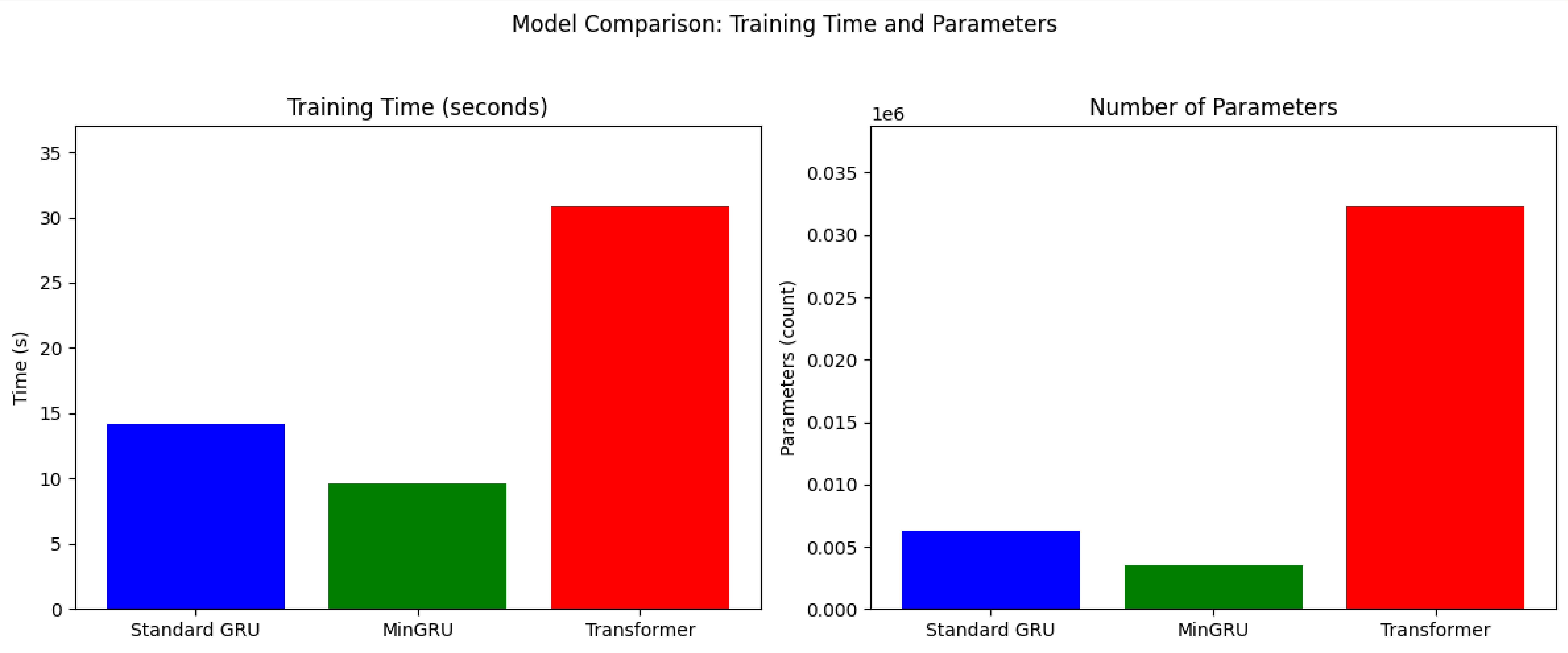
where a & b are independent on h(t-1)

This is similar to Parallel scan

```
h0  h1      h2  h3
  \  /        \  /
  h01       h23
     \     /
     h0123
```

TRAINING LOSS

GRU & MIN GRU &TRANSFORMER MODEL


Model Comparison: Training Time and Parameters
Training Time (seconds)
Time (s)
35
30
25
20
15
10
5
0
Standard GRU
MinGRU
Transformer
Number of Parameters
1e6
Parameters (count)
0.035
0.030
0.025
0.020
0.015
0.010
0.005
0.000
Standard GRU
MinGRU
Transformer

Thank you.