GRU (Gated Recurrent Unit)

- Goal: To capture long-term dependencies in sequences with fewer parameters and simpler structure than LSTM.

- Math:

A GRU uses a single hidden state h_t and controls information flow using two gates.

- The update gate decides how much past information to keep:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$

- The reset gate decides how much past information to forget:

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

- A candidate hidden state is computed:

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$$

- The final hidden state is:

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

- Conclusion:

GRU simplifies gated recurrence by merging memory and hidden state, making it faster to train while still handling long-term dependencies effectively.