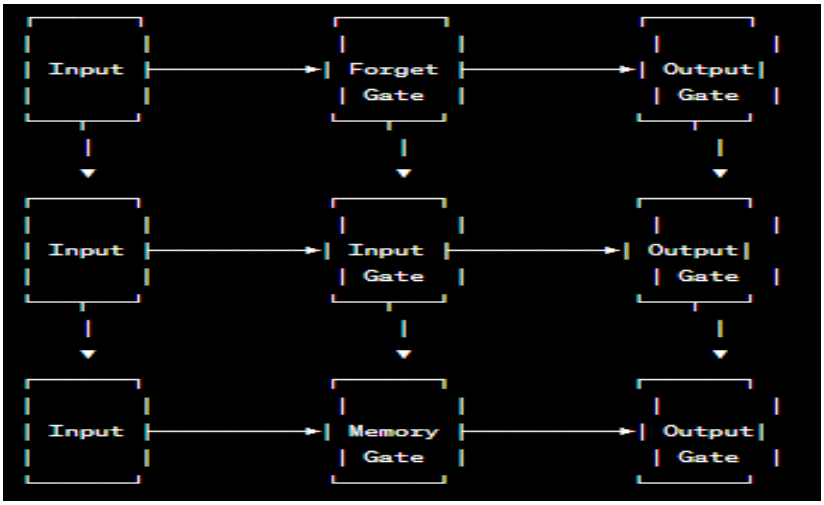
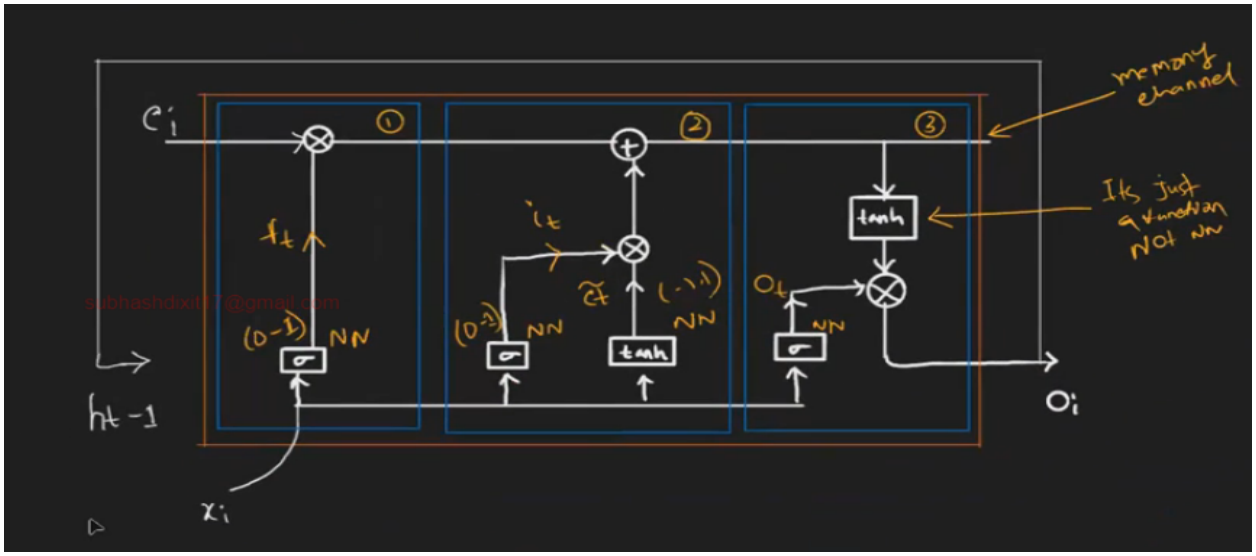
NLP: Day 4

LSTM - Long Short-Term Memory:

- LSTM is a type of recurrent neural network (RNN) architecture that addresses the issue of capturing long-term dependencies in sequential data.
- **Components of LSTM:**
  - Forget Gate: Controls which information from the previous cell state should be forgotten.
  - Input Gate / Memory Gate: Determines which information to store in the current cell state.
  - Output Gate: Determines how much information from the cell state should be exposed as the output.
- **Activation functions used in LSTM:**
  - Forget Gate: Sigmoid function (0 to 1)
  - Input Gate / Memory Gate: Hyperbolic tangent function (tanh) (-1 to 1)
  - Output Gate: Hyperbolic tangent function (tanh) (-1 to 1)

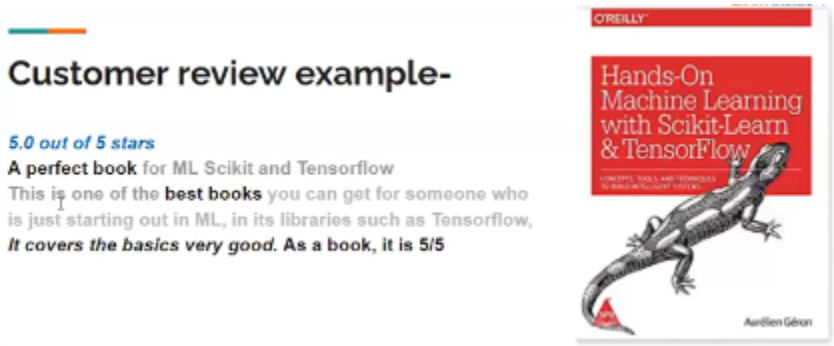
- LSTM can learn and retain context information within its memory cell, allowing it to capture long-term dependencies in the data.
- **Example:** Consider the sentence "My name is Subhash. Now, I am learning about LSTM." To understand the context of the phrase "Now, I am learning about LSTM", it is not necessary to have the previous sentence "My name is Subhash". LSTM can learn to focus on the relevant information.





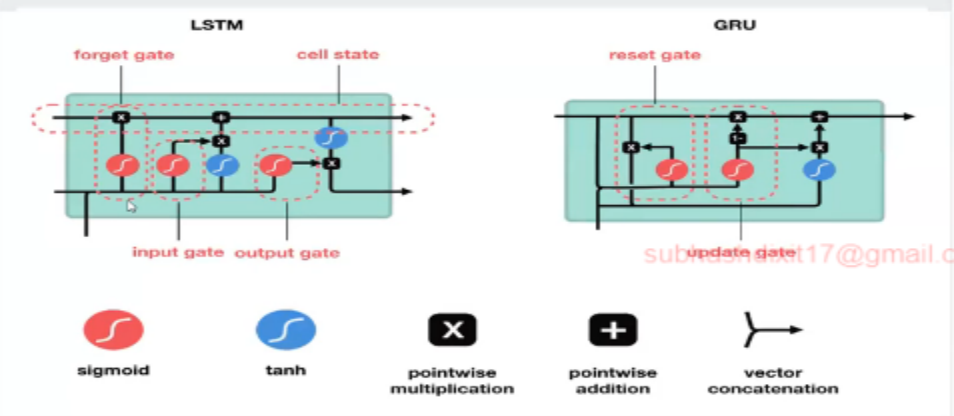
Xi = Input
Oi = Output
Ci = Memory Channel
ht - i = Input from previous output

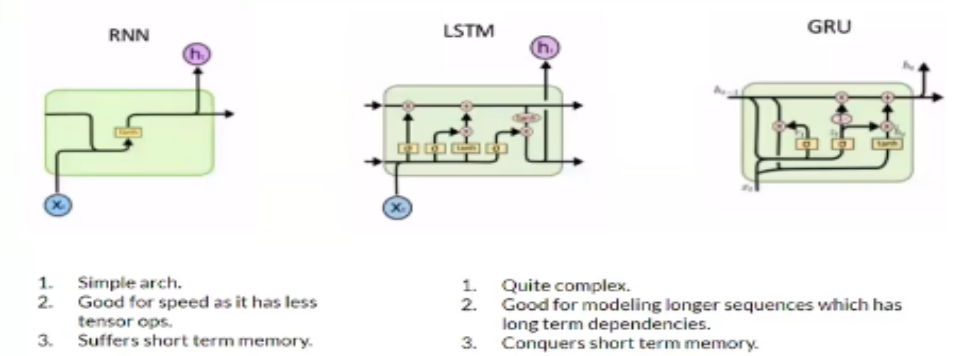
- To give sentiments, we need a few words, not the entire sentence. That's why LSTM is important



LSTM vs. GRU vs. RNN:

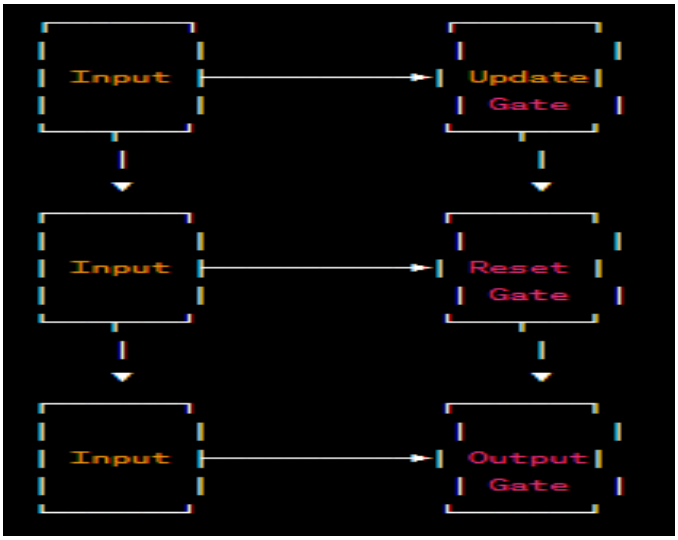
- LSTM and GRU are both variations of the RNN architecture designed to address the vanishing gradient problem and capture long-term dependencies.
- GRU (Gated Recurrent Unit) is a simplified version of LSTM, combining the forget and input gates into a single update gate.
- GRU does not have a separate memory channel but utilizes a different mechanism to store and update previous information.

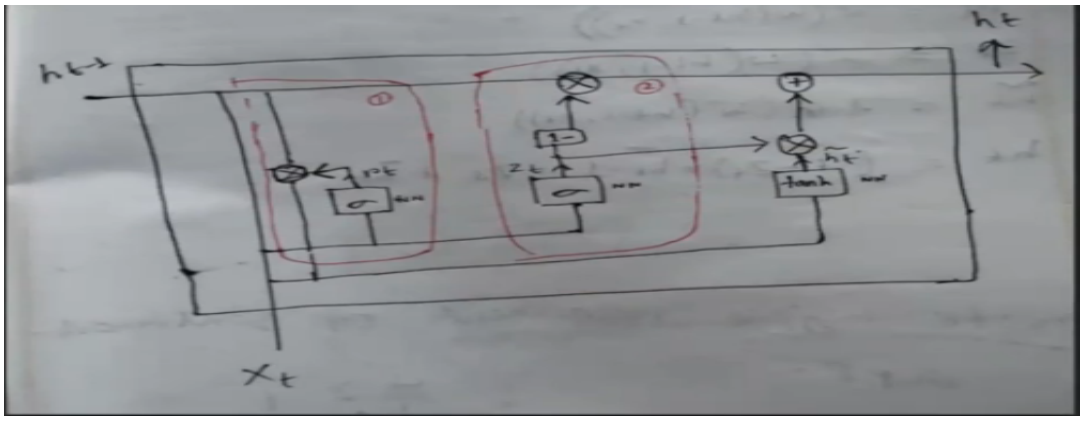




GRU (Gated Recurrent Unit):

- GRU is a type of recurrent neural network (RNN) architecture, similar to LSTM, that addresses the vanishing gradient problem and captures long-term dependencies in sequential data.
- **Components of GRU:**
  - Update Gate: Controls the amount of previous information to retain and update.
  - Reset Gate: Determines which information to discard from the previous state.
- **Activation functions used in GRU:**
  - Sigmoid function (0 to 1) for the update and reset gates.
  - Hyperbolic tangent function (tanh) (-1 to 1) for the candidate activation.
- GRU simplifies the LSTM architecture by combining the forget and input gates into a single update gate.
- **Example:** Consider the sentence "My name is Subhash. Now, I am learning about GRU." The GRU network can focus on the relevant information in the phrase "Now, I am learning about GRU" without needing the previous sentence "My name is Subhash".
- It doesn’t have any separate memory channel but it has some way to store previous information





Xt - Input
ht - i - previous output as input
"-1" - To store previous information