Gates, LSTM

Deep Neural Network

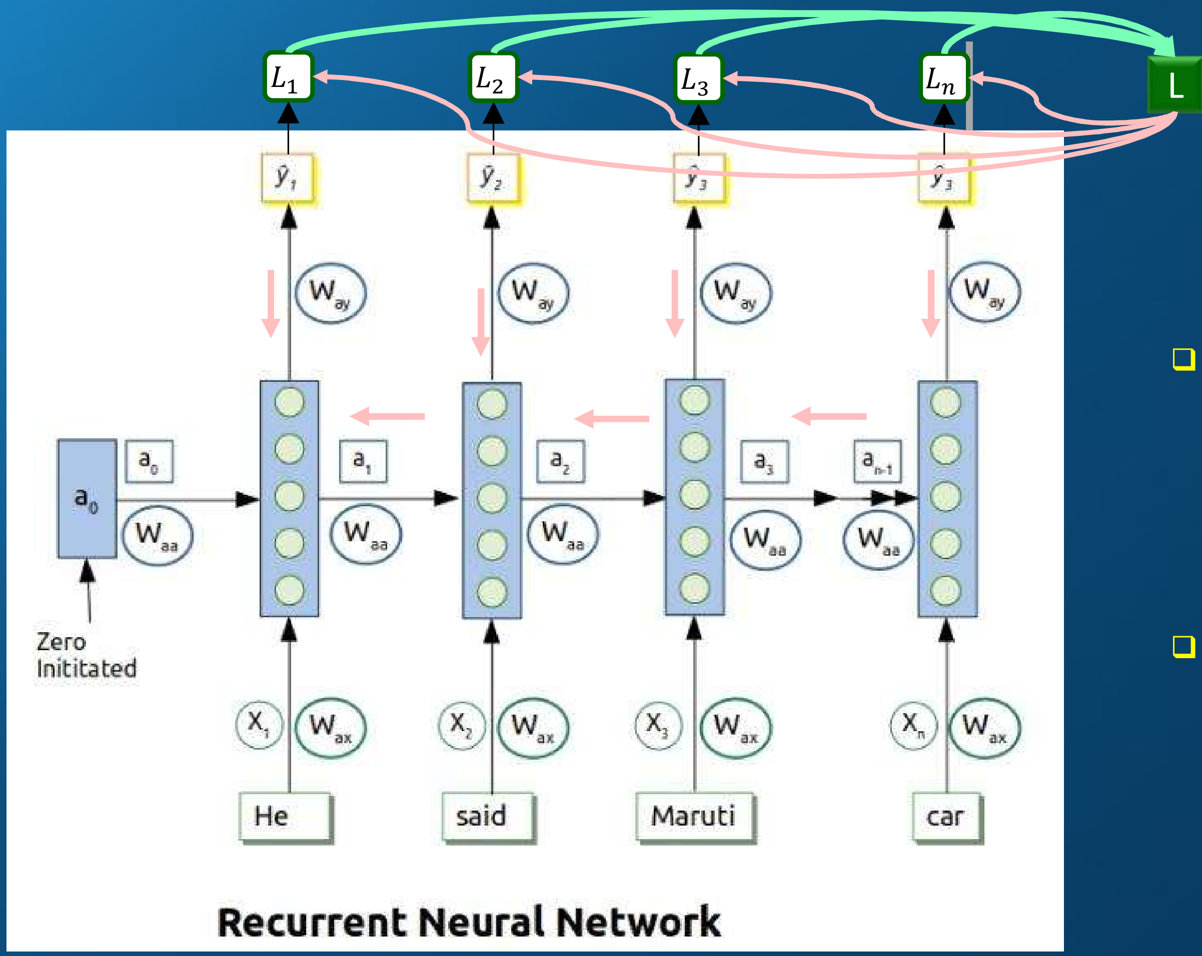
Session 20

Pramod Sharma
pramod.sharma@prasami.com

Agenda

- LSTM
- LSTM vs GRU
- Bidirectional RNN
- Putting all together – Deep RNN
- Attention Model

Back Propagation



Forword propagation:

- $a_t = g_1([a_{t-1}, x_t] \,.\, W_a + b_a)$
- $\hat{y}_t = g_2\big(a_t . W_y + b_y\big)$

- At time step ‘t’; Loss Function for single prediction
  - $L_t(\hat{y}_t, y) = - y_t \,.\, \log(\hat{y}_t) - (1 - y_t) \,.\, \log(1 - \hat{y}_t)$

- Sum of losses at all time steps:
  - $L(\hat{y}, y) = \sum_{t=1}^{T_x} L_t(\hat{y}_t, y)$

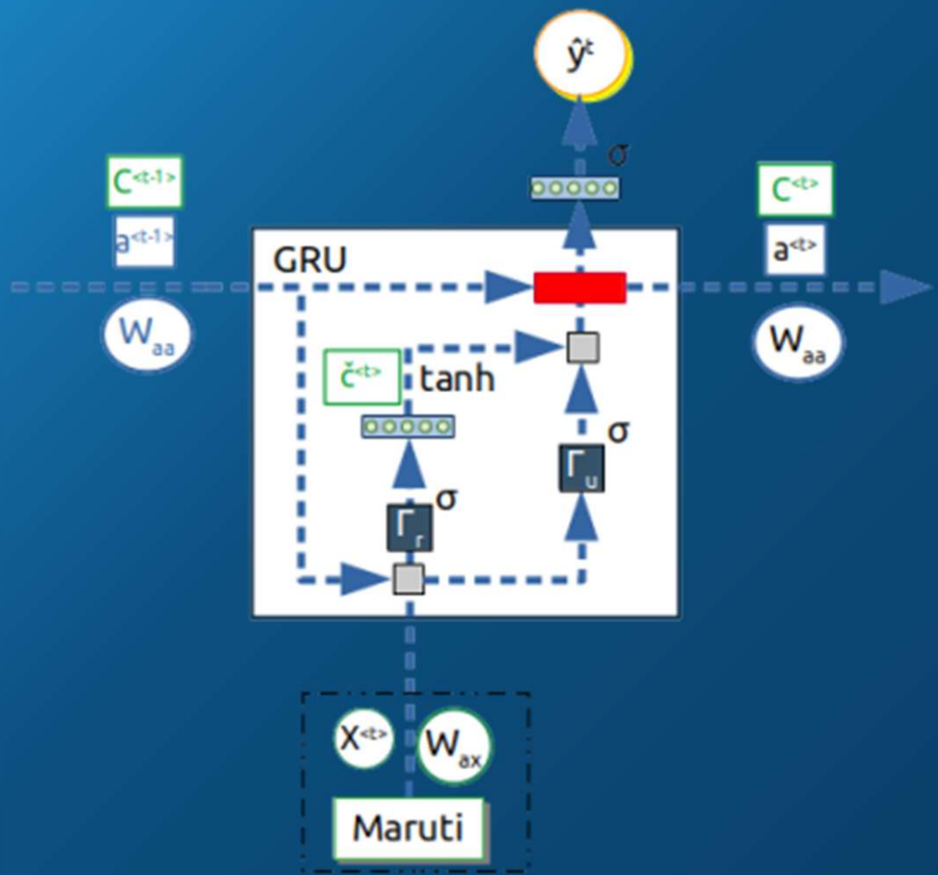
Long Short Term Memory network – LSTM

- A special kind of RNN, capable of learning long-term dependencies
- Introduced by Hochreiter & Schmidhuber (1997)
- Were refined and popularized by many people in following work
- LSTM were on a kind of back burner till 2013
- Original paper is quite mathematical and little overwhelming to follow
  - It goes into depths of Exploding and Vanishing Gradients
  - AI Community could not appreciate its value at that time

Long Short Term Memory network – LSTM

- LSTM work tremendously well on a large variety of problems, and are now widely used.
  - Speech recognition, Language modeling, Translation, Image captioning…
- LSTMs are explicitly designed to avoid the long-term dependency problem
- Designed to remember information for multiple time steps
- The key to LSTMs is the cell state
  - We have seen similar cell in GRU
- The cell state carry information through either unchanged or with updates

GRU Cell



- Recall our discussions on GRU

Extended GRU:

$\check{c}_t = \tanh([\Gamma_r * c_{t-1} : x_t] \cdot W_c + b_c)$

$\Gamma_u = \sigma([c_{t-1} : x_t] \cdot W_u + b_u)$

$\Gamma_r = \sigma([c_{t-1} : x_t] \cdot W_r + b_r)$

$c_t = \Gamma_u \cdot \check{c}_t + (1 - \Gamma_u) \cdot c_{t-1}$

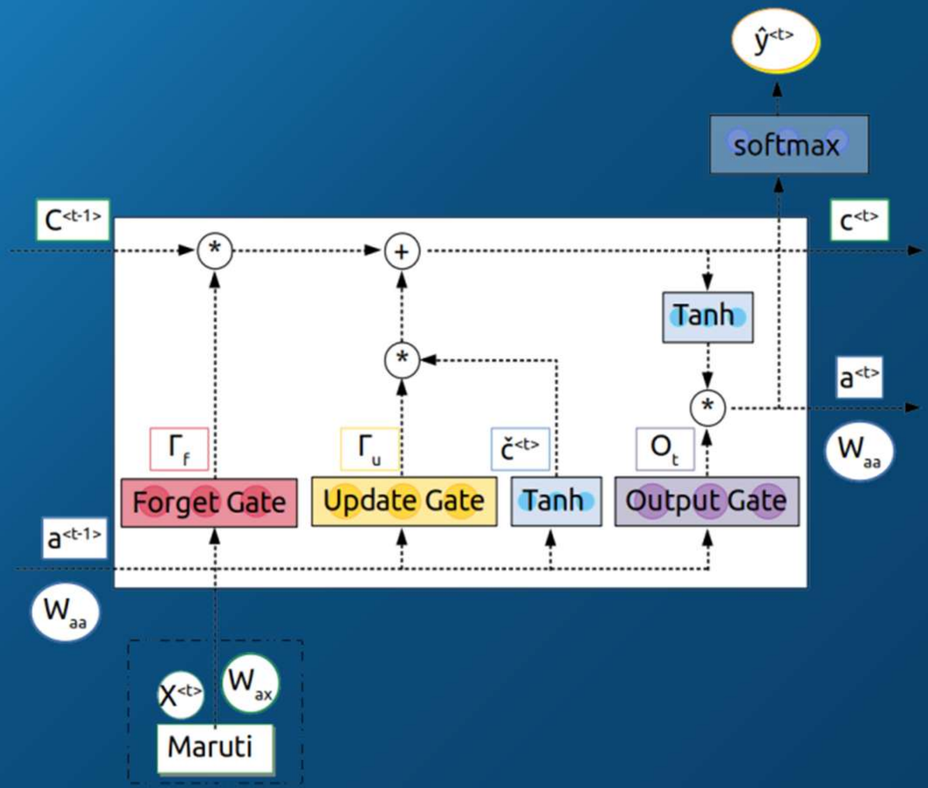
If $\Gamma_u = 1$ then c_t will be equal to $\check{c}_t$,
If $\Gamma_u = 0$ then c_t will be equal to c_{t-1}

And as usual $a_t = c_t$

Long Short Term Memory network – LSTM

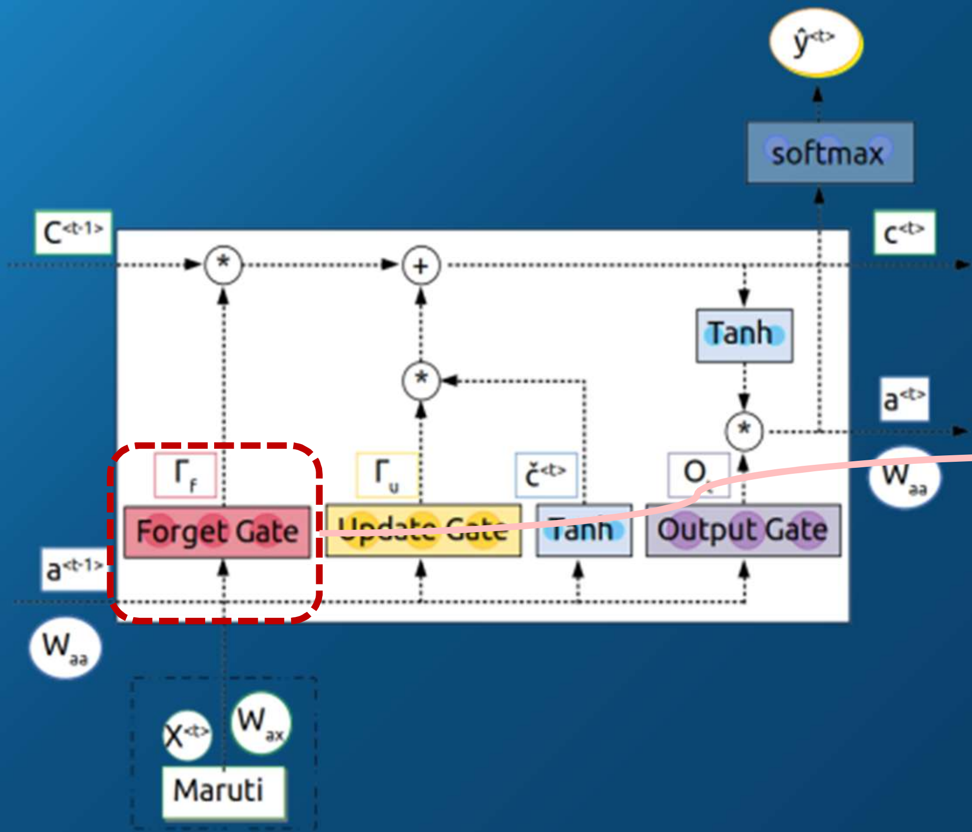
- Information can be removed or added to the cell state
- The structure regulating the information is called gates
- Gates are a way to optionally let information through or otherwise.
- Gates have sigmoid activation resulting in almost 0, 1 (all or nothing) kind of behavior

Overall



- Lets make a few changes in GRU Cell

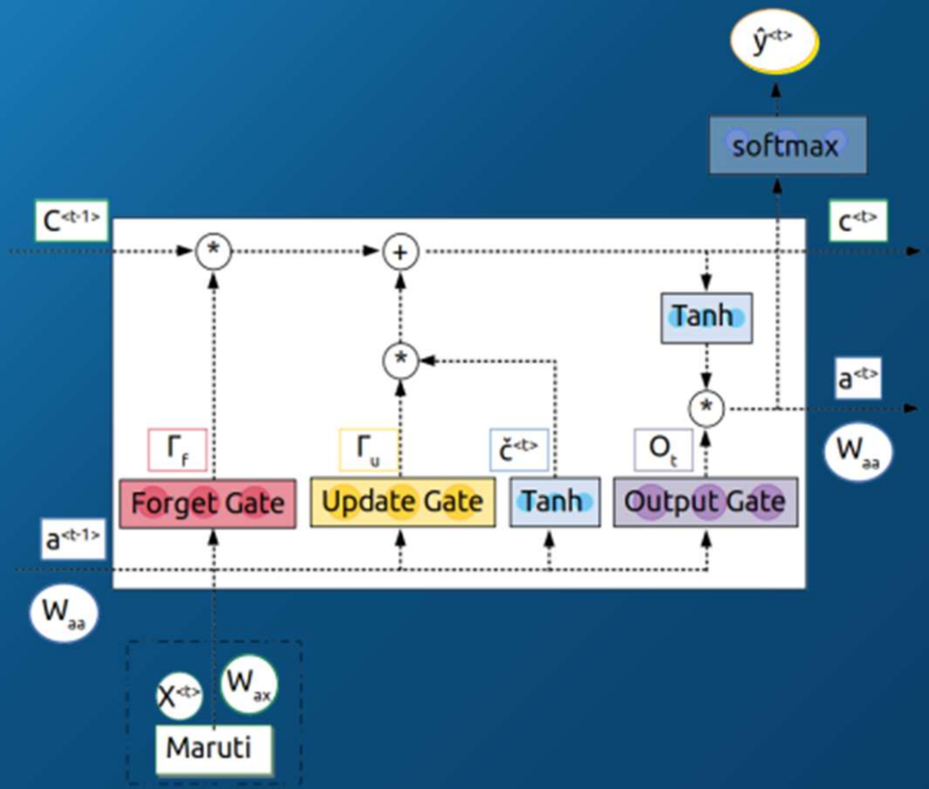
Forget Gate



- Lets make a few changes in GRU Cell
  - Equation $c_t = \Gamma_u * \hat{c}_t + (1 - \Gamma_u) * c_{t-1}$ is modified to
  - Equation $c_t = \Gamma_u * \hat{c}_t + \Gamma_f * c_{t-1}$
- Forget gate decides what information to throw away from the cell state
  - $\Gamma_f = \sigma([a_{t-1} : X_t] . W_f + b_f)$
- Forget gate value is between 0 and 1 depending upon a_{t-1} and X_t.
  - 1 represents “completely keep this”
  - 0 represents “completely get rid of this”
  - Or something “in-between”…

pra-sâmi

Forget Gate



Keep

I **felt happy** because I saw the others **were happy**

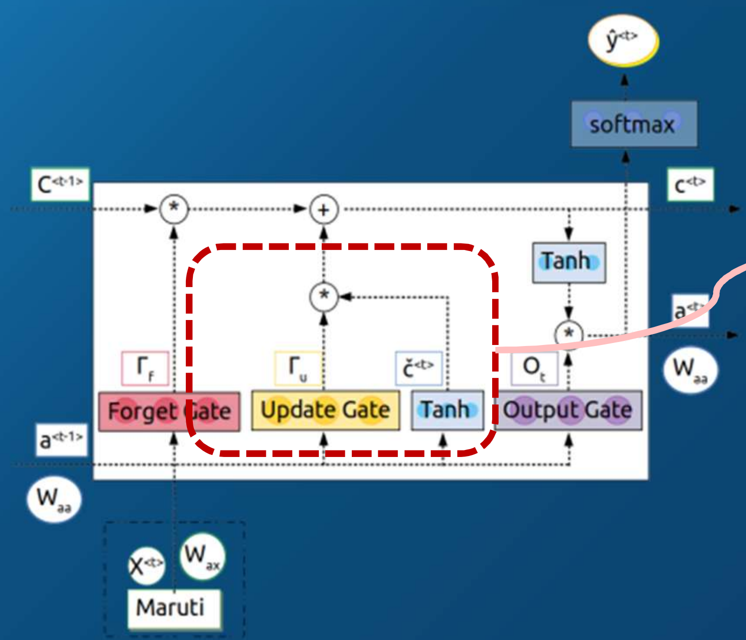
Keep

and because I knew I should **feel happy,** but I

Forget

wasn't really happy.

Update Gate



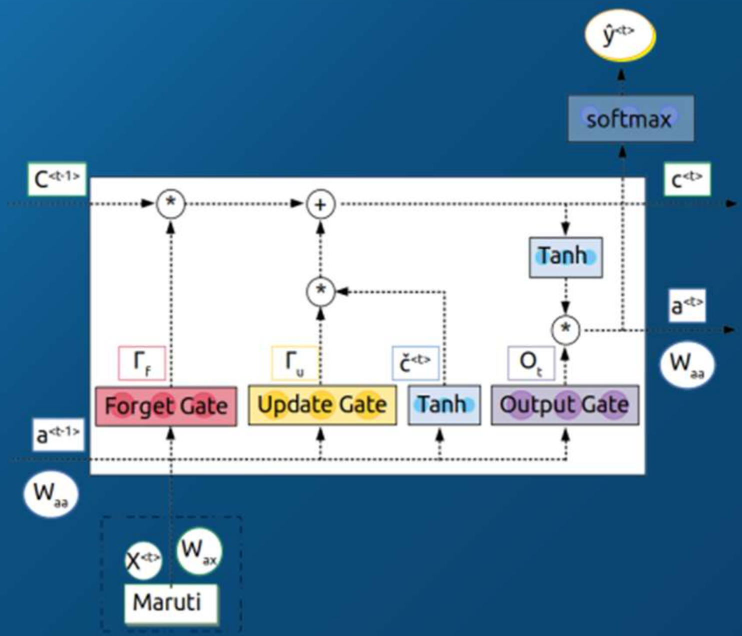
- What new information we're going to store in the cell state.

- Two step Process
  - First, a sigmoid layer called the "Update Gate" decides which values we'll update
  - $\Gamma_u = \sigma\ ([\ a_{t-1}\ : X_t\ ]\ . W_u + b_u)$

- Next, a tanh layer creates a vector of new candidate values, $\hat{c}_t$
  - $\hat{c}_t = \tanh\ ([a_{t-1}\ : X_t]\ . W_c + b_c)$

- Next step, combine these two to create an update to the state.
  - $c_t = \Gamma_u * \hat{c}_t + \Gamma_f * c_{t-1}$

Update Gate



New, update **Keep**

I **felt happy** because I saw the others **were happy**

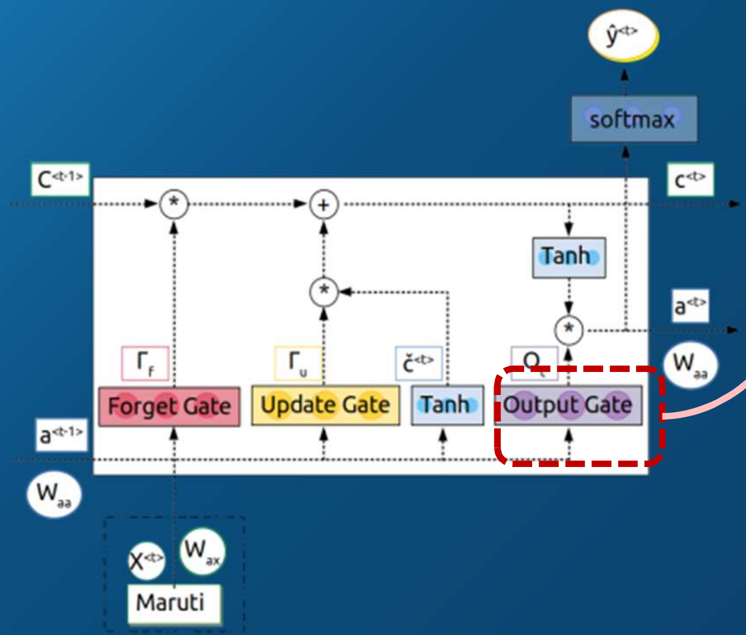
Keep

and because I knew I should **feel happy,** but I

forget, update

wasn't really happy.

Output Gate

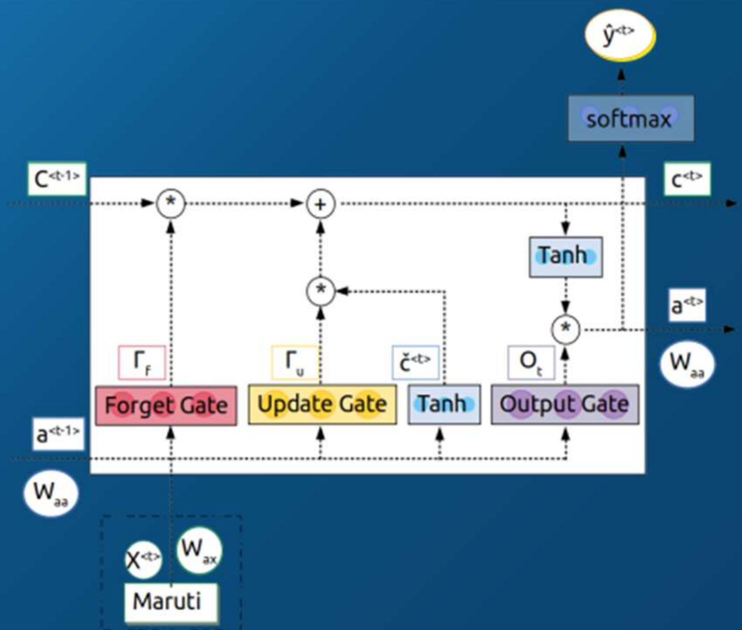


- What's output.
- Two step Process
  - First, we run a sigmoid layer which decides what parts of the cell state we're going to output.
  - Γ_o = σ ([a_{t-1} : X_t] . W_o + b_o)
  - Next, a process c_t through tanh activation and multiply by Γ_o
  - a_t = Γ_o * tanh (c_t)
- We can also use a_t to calculate $\hat{y}_t$

pra-sâmi

14

Output Gate



Positive **Positive**

I **felt happy** because I saw the others **were happy**

Positive

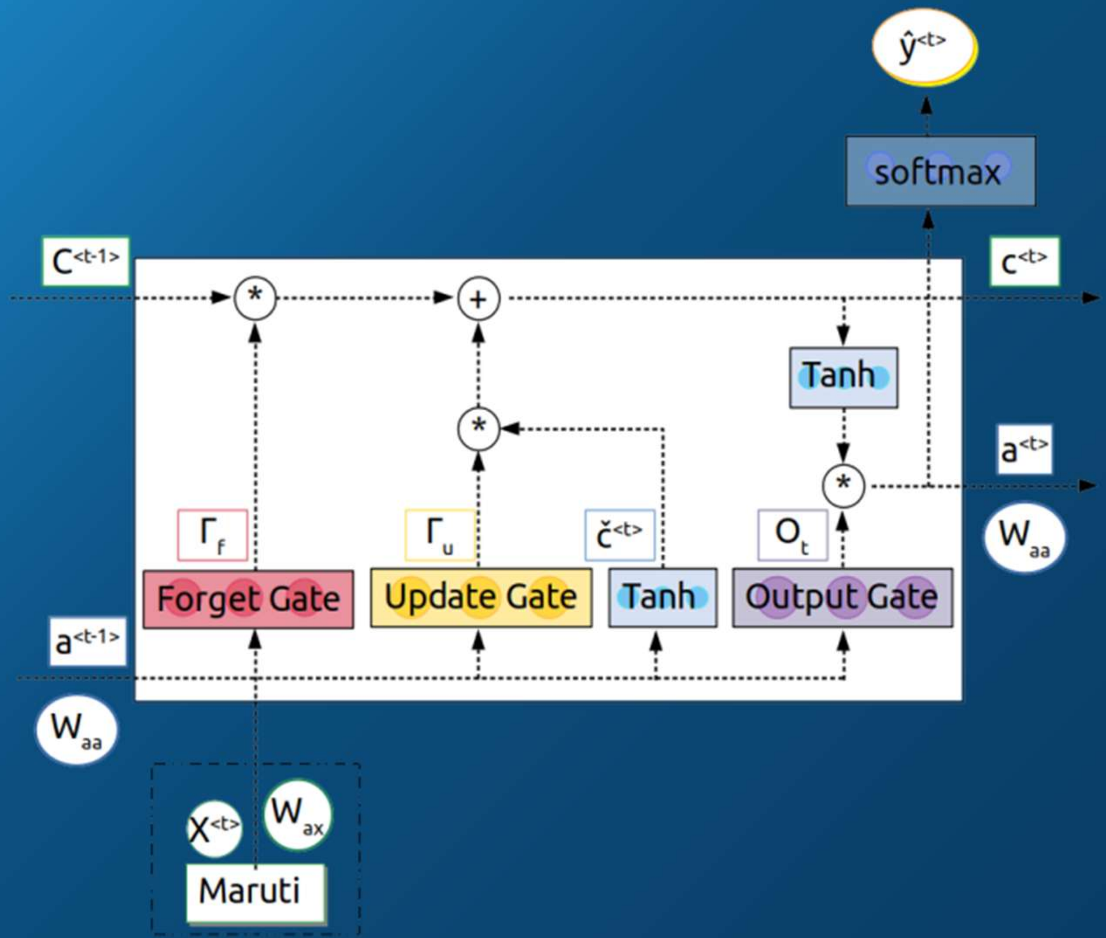
and because I knew I should **feel happy,** but I

Negative Review

wasn't really happy.

pra-sâmi

Overall



$\hat{c}_t = \tanh\left([a_{t-1} : X_t] \,.\, W_c + b_c \right)$

$\Gamma_u = \sigma\left([a_{t-1} : X_t] \,.\, W_u + b_u \right)$

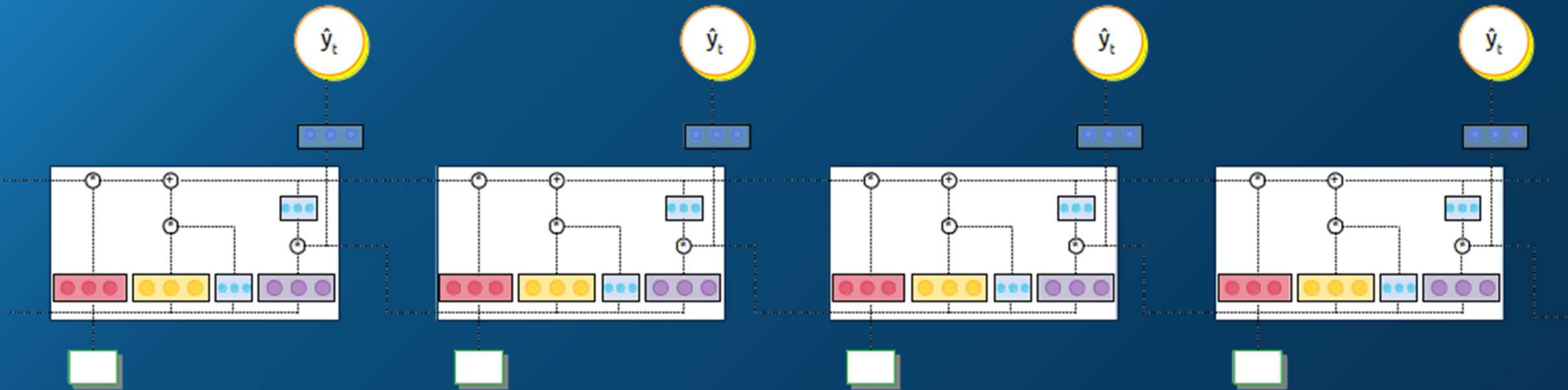
$\Gamma_f = \sigma\left([a_{t-1} : X_t] \,.\, W_f + b_f \right)$

$\Gamma_o = \sigma\left([a_{t-1} : X_t] \,.\, W_o + b_o \right)$

$c_t = \Gamma_u * \hat{c}_t + \Gamma_f * c_{t-1}$

$a_t = \Gamma_o * \tanh(c_t)$

Chain of LSTM cell…



Variants of LSTM

- Almost every other paper comes out with some variant of LSTM
- LSTM variant, introduced by Gers & Schmidhuber (2000),
  - Adding “peephole connections.”
  - Let the gate layers look at the cell state.

$\hat{c}_t = \tanh\ ([a_{t-1}\ : X_t : c_{t-1}\ ]\ . W_c + b_c)$

$\Gamma u = \sigma\ ([a_{t-1}\ : X_t : c_{t-1}\ ]\ . W_u + b_u)$

$\Gamma f = \sigma\ ([a_{t-1}\ : X_t : c_{t-1}\ ]\ . W_f + b_f)$

$\Gamma o = \sigma\ ([a_{t-1}\ : X_t : c_{t-1}\ ]\ . W_o + b_o)$

$c_t = \Gamma_u * \hat{c}_t + \Gamma_f * c_{t-1}$

$a_t = \Gamma_o * \tanh\ (c_t\ )$

$\hat{c}_t = \tanh\ ([a_{t-1}\ : X_t\ ]\ . W_c + b_c)$

$\Gamma_u = \sigma\ ([\ a_{t-1}\ : X_t\ ]\ . W_u + b_u)$

$\Gamma_f = \sigma\ ([\ a_{t-1}\ : X_t\ ]\ . W_f + b_f)$

$\Gamma_o = \sigma\ ([\ a_{t-1}\ : X_t\ ]\ . W_o + b_o\ )$

$c_t = \Gamma_u * \hat{c}_t + \Gamma_f * c_{t-1}$

$a_t = \Gamma_o * \tanh\ (c_t\ )$

- You have already seen other most popular variant GRU

LSTM vs GRU

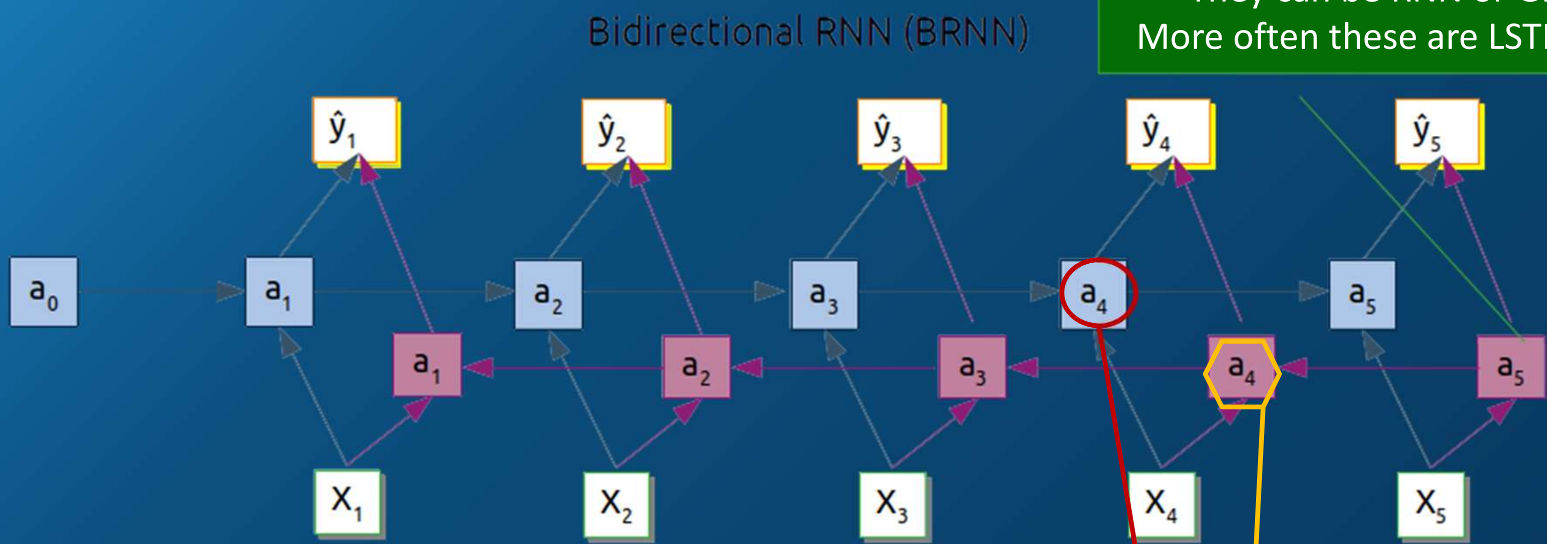
LSTM vs GRU

- Different Problems, different algorithms work
- NO clear choices
- In general, GRU is faster
- Try both and see which one produces better results.

Bidirectional RNN

Bidirectional RNN

They can be RNN or GRU or LSTM blocks
More often these are LSTM blocks in the BRNN

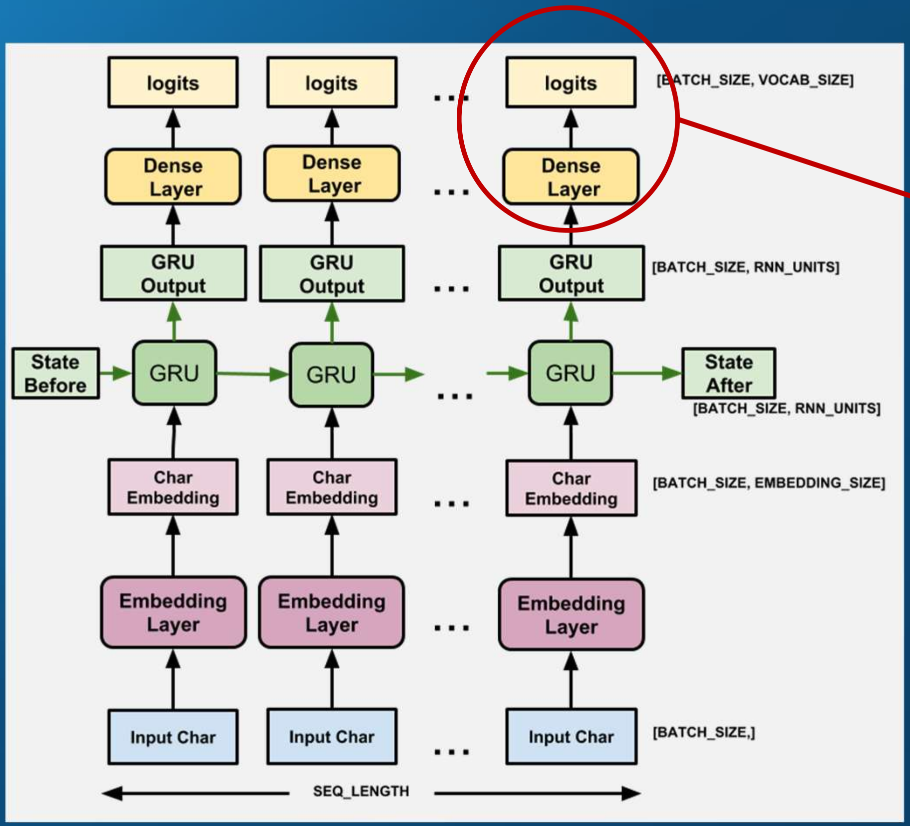


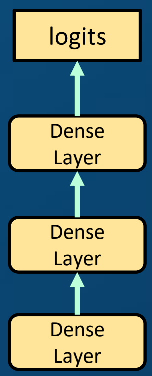
- "**He said Maruti is** most fuel efficient"
- "**He said Maruti is** most expensive shop"
- "**He said Maruti is** strongest"

- $\hat{y}_l = g([\ a_l : a_l\ ] . W_y + b_y)$
- One limitation: you need complete sentences before any predictions. May not work for voice translation as we need the dialog to finish which can be way out…

Putting all together – Deep RNN

Putting it together…





- You may see multiple dense layers without horizontal connection
- Its rare to see more than 3 GRU or LSTM units stacked up vertically… Network is already too big!

Attention Model

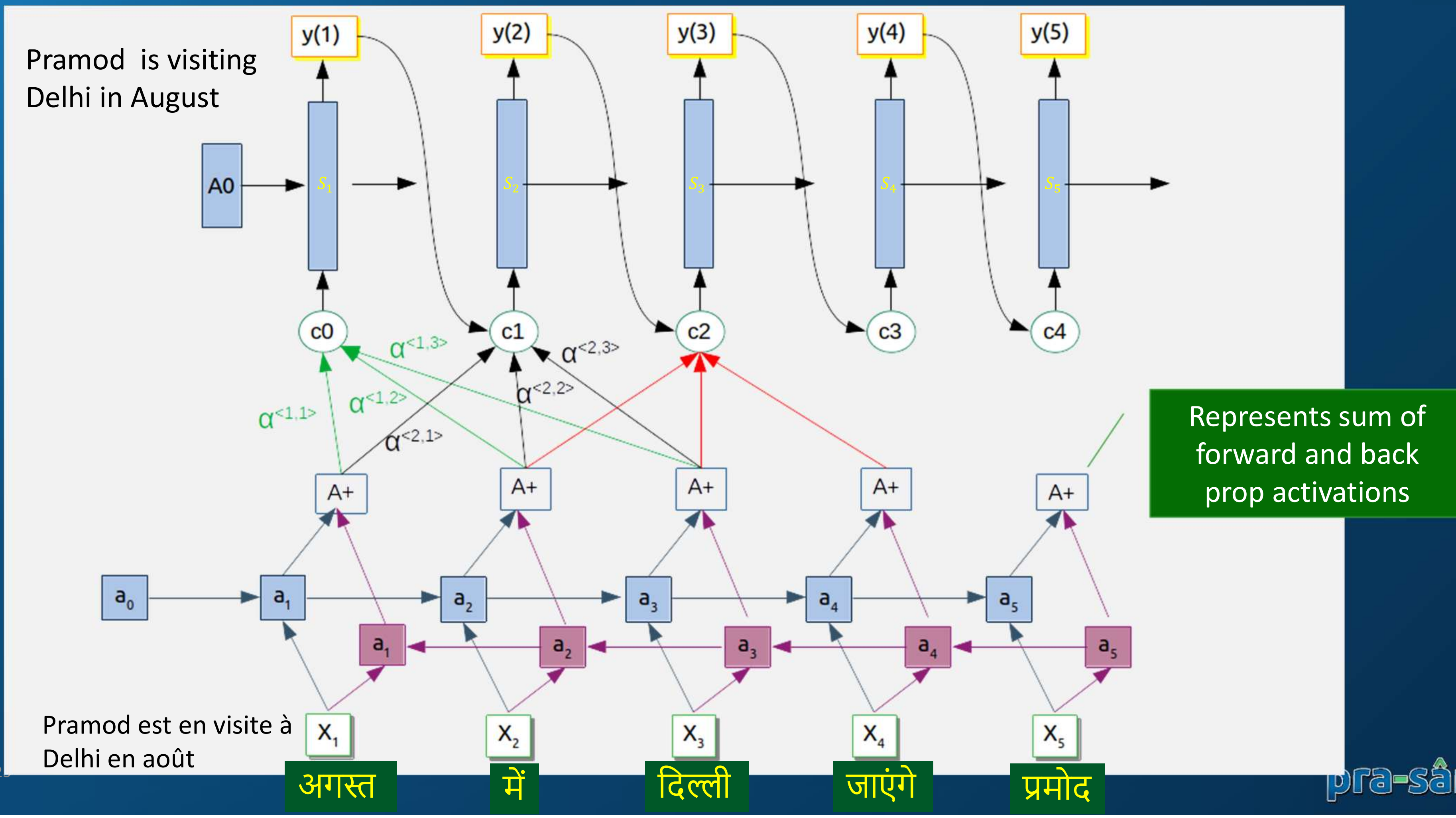
Given a very long sentence

- “As he crossed toward the pharmacy at the corner he involuntarily turned his head because of a burst of light that had ricocheted from his temple, and saw, with that quick smile with which we greet a rainbow or a rose, a blindingly white parallelogram of sky being unloaded from the van—a dresser with mirrors across which, as across a cinema screen, passed a flawlessly clear reflection of boughs sliding and swaying not arboreally, but with a human vacillation, produced by the nature of those who were carrying this sky, these boughs, this gliding façade.”

How would a human being would translate???????

Attention Model

Attention Model



pra-sâmi

Reflect…

- What is the key difference between Recurrent Neural Networks (RNNs) and Feedforward Neural Networks?
  - A) RNNs use activation functions
  - B) RNNs have cycles and feedback loops in their connections
  - C) RNNs cannot handle sequential data
  - D) RNNs use more neurons than Feedforward Networks
- Answer: B) RNNs have cycles and feedback loops in their connections
- Which of the following is a typical application of RNNs?
  - A) Image classification
  - B) Sentiment analysis of text
  - C) Object detection
  - D) Image segmentation
- Answer: B) Sentiment analysis of text
- What type of data is RNN most suitable for?
  - A) Tabular data
  - B) Sequential data like time series or text
  - C) Randomly ordered data
  - D) Static data like images
- Answer: B) Sequential data like time series or text
- What is the problem of vanishing gradients in RNNs?
  - A) The model grows too large over time
  - B) The gradients used for backpropagation become very small, making learning slow or ineffective
  - C) The model loses information about long sequences
  - D) The model overfits due to too much data
- Answer: B) The gradients used for backpropagation become very small, making learning slow or ineffective

Reflect…

- Which of the following is an RNN variant designed for processing sequences in both directions?
  - A) Unidirectional RNN
  - B) Bidirectional RNN
  - C) Convolutional Neural Network (CNN)
  - D) Autoencoder
- Answer: B) Bidirectional RNN
- What are the three types of gates used in LSTM networks?
  - A) Input gate, Output gate, Forget gate
  - B) Memory gate, Activation gate, Forget gate
  - C) Input gate, Reset gate, Output gate
  - D) Update gate, Forget gate, Reset gate
- Answer: A) Input gate, Output gate, Forget gate
- Why are Gated Recurrent Units (GRUs) considered simpler than LSTMs?
  - A) GRUs use a single gate instead of multiple gates
  - B) GRUs combine the forget and input gates into a single update gate
  - C) GRUs do not have a hidden state
  - D) GRUs do not require backpropagation
- Answer: B) GRUs combine the forget and input gates into a single update gate
- Which loss function is commonly used when training RNNs for sequence-to-sequence tasks like translation?
  - A) Mean Squared Error (MSE)
  - B) Cross-Entropy Loss
  - C) Hinge Loss
  - D) Triplet Loss
- Answer: B) Cross-Entropy Loss

Reflect…

- Which of the following architecture change is used to solve the vanishing gradient problem in RNNs?
  - A) Weight initialization
  - B) Long Short-Term Memory (LSTM) networks
  - C) Gradient clipping
  - D) Data augmentation

- Answer: B) Long Short-Term Memory (LSTM) networks

- In an RNN, how does the hidden state affect the model?
  - A) It acts as the output of the network
  - B) It serves as a temporary memory to retain information from previous time steps
  - C) It stores the weights of the model
  - D) It determines the learning rate of the model

- Answer: B) It serves as a temporary memory to retain information from previous time steps

  - Which component of the RNN is responsible for learning long-term dependencies?
  - A) Activation function
  - B) Bias term
  - C) Hidden state
  - D) Cell state (in LSTMs)

- Answer: D) Cell state (in LSTMs)

- What is the difference between an LSTM and a traditional RNN?
  - A) LSTMs have multiple layers of neurons
  - B) LSTMs use gates to control the flow of information and avoid vanishing gradients
  - C) LSTMs use feedforward connections
  - D) LSTMs are slower to train than RNNs

- Answer: B) LSTMs use gates to control the flow of information and avoid vanishing gradients

Reflect…

- What is the primary difference between GRUs and LSTMs?
  - A) GRUs have fewer gates than LSTMs
  - B) GRUs do not have any gates
  - C) GRUs are only used for text data
  - D) GRUs are slower to train than LSTMs
- Answer: A) GRUs have fewer gates than LSTMs
- Which of the following gates is unique to the GRU architecture?
  - A) Forget Gate
  - B) Relevance Gate
  - C) Input Gate
  - D) Output Gate
- Answer: B) Relevance Gate

- What is a key advantage of using GRUs over traditional RNNs?
  - A) GRUs do not suffer from the vanishing gradient problem
  - B) GRUs are computationally less expensive than LSTMs
  - C) GRUs have a better ability to learn from very short sequences
  - D) GRUs use convolutional layers for better feature extraction
- Answer: A) GRUs do not suffer from the vanishing gradient problem
- Which of the following functions do GRUs use to manage memory and control information flow?
  - A) Only the forget gate
  - B) Only the output gate
  - C) Relevance and Update gates
  - D) Relevance and Output gates
- Answer: C) Relevance and Update gates

Reflect…

- What is the key feature of LSTM networks compared to traditional RNNs?
  - A) LSTMs do not use any gates
  - B) LSTMs have a memory cell that can maintain information over long time periods
  - C) LSTMs are faster to train than RNNs
  - D) LSTMs are used only for image processing tasks
- Answer: B) LSTMs have a memory cell that can maintain information over long time periods
- Which of the following gates in an LSTM controls what portion of the past memory should be retained?
  - A) Input Gate
  - B) Forget Gate
  - C) Output Gate
  - D) Relevance Gate
- Answer: B) Forget Gate
- What is the role of the Input Gate in an LSTM?
  - A) To discard irrelevant information from the previous time step
  - B) To add new information to the cell state from the current input
  - C) To control the final output of the LSTM
  - D) To reset the hidden state
- Answer: B) To add new information to the cell state from the current input
- What allows LSTMs to mitigate the vanishing gradient problem?
  - A) The use of ReLU activation functions
  - B) The use of multiple hidden layers
  - C) The cell state, which allows the gradient to flow unchanged across time steps
  - D) The dropout regularization technique
- Answer: C) The cell state, which allows the gradient to flow unchanged across time steps


THANK YOU