ASSIGNMENT - 3

1. Explain the basic architecture of RNN cell.

Ans: 1. Basic RNN Cell Architecture:

* Recurrent Unit: The core of an RNN is the recurrent unit, often referred to as a cell. This unit processes an input vector (x\_t) at the current time step (t) and the hidden state (h\_(t-1)) from the previous time step. It combines this information to produce a new hidden state (h\_t) and an output (y\_t).
* Shared Parameters: RNNs share the same set of parameters (weight matrices and biases) across all time steps. This allows them to learn generic patterns across sequences. The weight matrices typically include:
  + U: governs connections from the input layer to the hidden layer.
  + W: governs connections within the hidden layer, capturing the recurrent nature.
  + V: governs connections from the hidden layer to the output layer.
* Activation Function: A non-linear activation function (e.g., tanh, ReLU) is applied to the hidden state computation to introduce non-linearity and enable the network to learn complex relationships.

2. Explain Backpropagation through time (BPTT)

Ans: BPTT is a specialized backpropagation algorithm used for training RNNs. It unfolds the RNN across all time steps, creating a chain-like structure where the output at each step depends on the inputs and hidden states of all previous steps. The error is then propagated backward through this unfolded network, allowing the network to learn and adjust its parameters based on the sequence information. However, BPTT can suffer from vanishing and exploding gradients, which limit its effectiveness for longer sequences.

3. Explain Vanishing and exploding gradients

Ans: Vanishing and Exploding Gradients:

* Vanishing Gradients: As the error is backpropagated through many time steps in an RNN, the gradients can become very small or vanish altogether. This makes it difficult for the network to learn long-term dependencies.
* Exploding Gradients: Conversely, gradients can sometimes explode, causing the network weights to update too drastically and hindering convergence during training.

4. Explain Long short-term memory (LSTM)

Ans: LSTM is a special type of RNN cell designed to address the vanishing and exploding gradient problems. It introduces three gates (forget gate, input gate, output gate) that control the flow of information within the cell:

* Forget Gate: Decides which information to retain from the previous cell state (c\_(t-1)) based on the current input (x\_t) and the previous hidden state (h\_(t-1)).
* Input Gate: Determines which new information to store in the cell state (c\_t).
* Output Gate: Controls what information from the cell state (c\_t) is reflected in the output (y\_t).

5. Explain Gated recurrent unit (GRU)

Ans: GRU is another variant of RNN that addresses vanishing gradients while being computationally simpler than LSTMs. It combines the functionality of the forget gate and input gate of LSTMs into a single update gate, making it more efficient.

6. Explain Peephole LSTM

Ans: Peephole LSTM is an extension to LSTMs where the gates also have access to the cell state (c\_t) directly. This allows for better control of information flow and can sometimes improve performance on specific tasks.

7. Bidirectional RNNs

Ans: Bidirectional RNNs process sequences in both forward and backward directions. This allows the network to capture dependencies from both the past and future context, which can be beneficial for tasks like machine translation and sentiment analysis. There are two main ways to implement Bidirectional RNNs:

* Concatenated RNNs: Two separate RNNs are used, one processing the sequence forward and the other processing it backward. Their outputs are then concatenated at each time step.
* LSTM with Bidirectional Layers: A single LSTM layer is used, but the hidden state at each time step combines information from both the forward and backward passes.

8. Explain the gates of LSTM with equations.

Ans: Gates of LSTM with Equations (continued):

* Cell State (c\_t):
  + c\_t = f\_t \* c\_(t-1) + i\_t \* C\_t^~
* Output Gate (o\_t):
  + o\_t = σ(W\_o \* [x\_t, h\_(t-1)] + b\_o)
* Hidden State (h\_t):
  + h\_t = o\_t \* tanh(c\_t)

Here's a breakdown of the symbols:

* x\_t: Current input vector at time step t.
* h\_(t-1): Hidden state from the previous time step (t-1).
* f\_t, i\_t, o\_t: Forget gate, input gate, and output gate values at time step t, respectively (all between 0 and 1 due to the sigmoid activation function, σ).
* C\_t^~: Candidate cell state at time step t.
* c\_t: Cell state at time step t.
* W\_f, W\_i, W\_c, W\_o: Weight matrices for the forget gate, input gate, candidate cell state, and output gate, respectively.
* b\_f, b\_i, b\_c, b\_o: Biases for the forget gate, input gate, candidate cell state, and output gate, respectively.

9. Explain BiLSTM

Ans: BiLSTM is a variant of LSTM that processes sequences in both forward and backward directions. It achieves this by:

* Two LSTMs: It uses two separate LSTM layers. One processes the sequence forward, and the other processes it backward.
* Concatenated Hidden States: At each time step, the hidden states from both LSTMs are concatenated to create a richer representation that captures information from both the past and future context. This combined hidden state can then be used for prediction or further processing.

10. Explain BiGRU

Ans: BiGRU is a bidirectional version of the Gated Recurrent Unit (GRU). It employs two separate GRU layers, one processing forward and the other processing backward. At each time step, the hidden states from these GRUs are concatenated to create a combined representation for prediction or further processing.