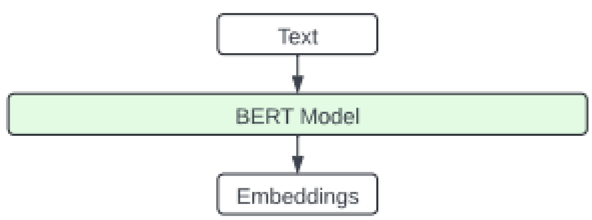
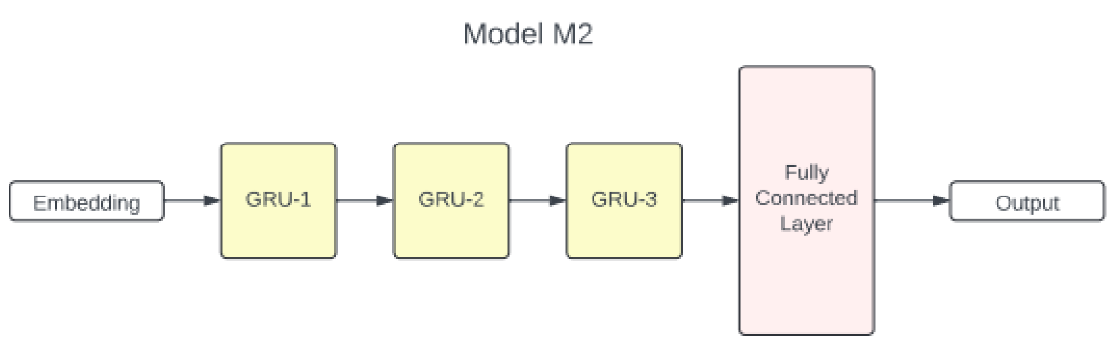
Assignment 4

Embedding- BERT embeddings were used along with padding,with batch size 32 and max length 25.

Model Architecture:



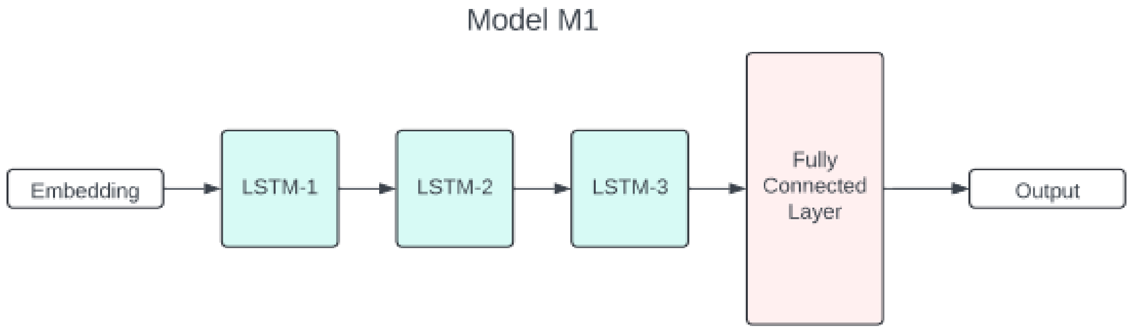
1. GRU Model Structure



Layer Configuration:

- First GRU layer with an input from the embedding dimension converting to 256 hidden units.
- Second GRU layer reduces the hidden units from 256 to 64.
- Third GRU layer further reduces the units from 64 to 16.
- A final fully connected layer that maps the 16 units to the number of classes for the prediction output.

2. LSTM Model Structure



Layer Configuration:

- First LSTM layer with an input from the embedding dimension converting to 256 hidden units.
- Second LSTM layer reduces the hidden units from 256 to 64.
- Third LSTM layer further reduces the units from 64 to 16.
- A final fully connected layer that maps the 16 units to the number of classes.

Evaluation and Results

After training both models for 80 epochs on the dataset, the following F1 scores were obtained for the emotion recognition task:

1. **LSTM Model:**
   Val Macro F1-Score: 0.6203351842352914
   Number of flips detected by model - 1103
   Total number of flips in the data - 3132
   Percentage of flips detected - 35.22 %
2. **GRU Model:**
   Val Macro F1-Score: 0.7412858709165464
   Number of flips detected by model - 1609
   Total number of flips in the data - 3132
   Percentage of flips detected - 51.37 %

From these results, the GRU model outperforms the LSTM model significantly with a higher F1 score.

Analysis
Advantages of GRU Over LSTM
- Simplicity and Efficiency: GRU has a simpler structure with fewer parameters due to its gating mechanism, which can make it computationally more efficient and faster to train than LSTM.
- Faster Convergence: In practice, GRUs tend to converge faster, which can be crucial for training efficiency, especially on datasets with complex and varied emotional contexts.
- Handling of Vanishing Gradient Problem: Both GRU and LSTM are designed to combat the vanishing gradient problem in long sequences. However, the GRU's simpler gating mechanism might have provided an edge in learning long-term dependencies more effectively in this context.

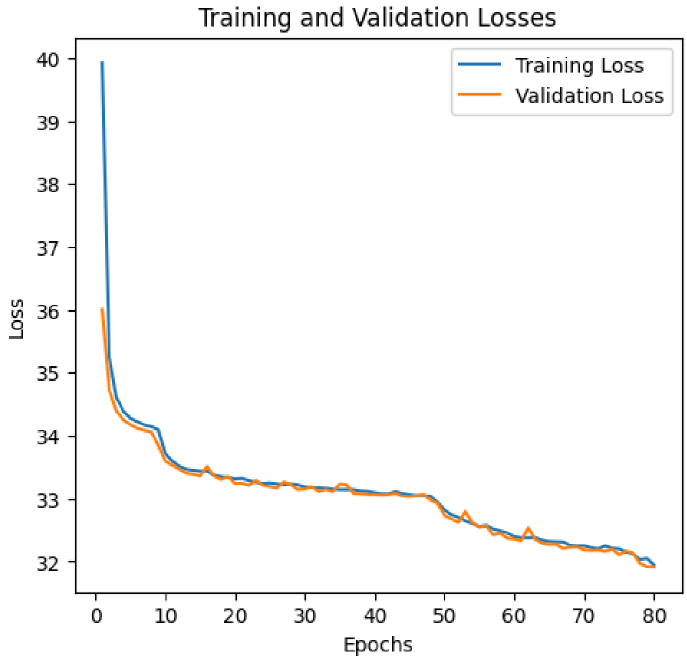
Reason for LSTM's Lower Performance
- Complexity: LSTMs contain more parameters due to an additional gate (the output gate), which increases the computational burden and may lead to slower learning or overfitting in some cases.

Conclusion

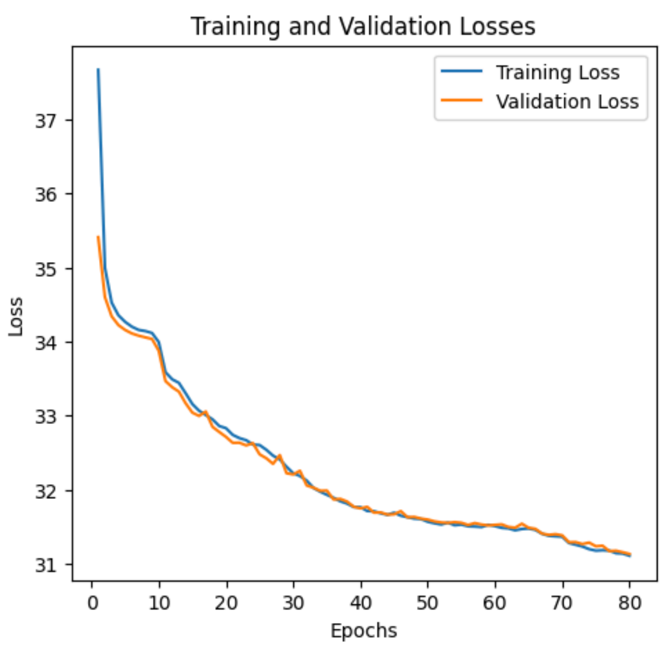
The GRU model exhibited superior performance over the LSTM model in emotion recognition within conversational contexts using the MELD dataset. The GRU’s architectural efficiency, quicker training times, and robustness in learning dependencies make it preferable for this specific application.

Intuition: The LSTM model's sequential learning capability allows it to capture the dynamics of emotional shifts in conversations, where emotions can change rapidly and unpredictably. The use of GRU (Gated Recurrent Unit) models in the context of ERC is crucial due to their ability to handle sequences, such as text or speech, which are inherent in conversational data. Also, the experience of training and testing these models in the previous assignments

Train and Val loss plots for M1 (GRU)-



Train and Val loss plots for M2 (LSTM)-

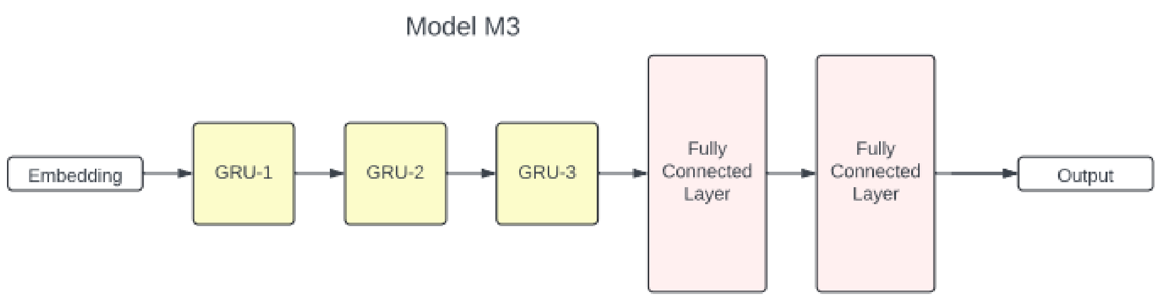


Model Architectures

Embedding- BERT embeddings were used along with padding,with batch size 32 and max length 25.

Model Descriptions

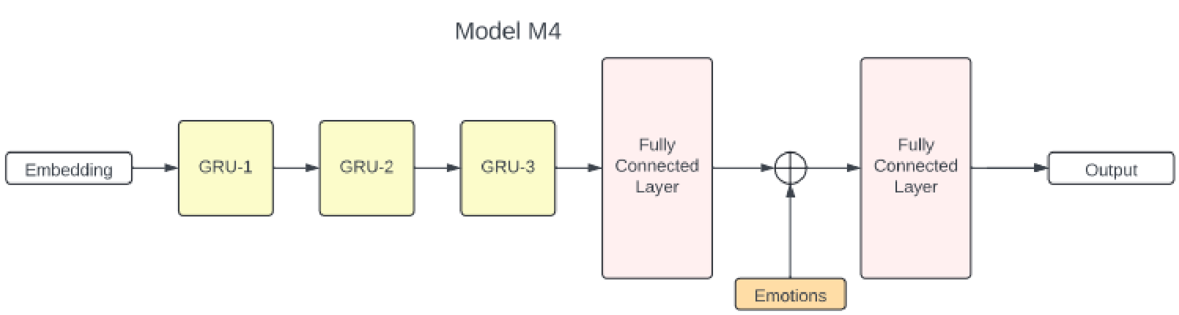
1. GRUModel



Architecture: This model is built using a sequence of Gated Recurrent Unit (GRU) layers with decreasing hidden units (256, 64, and 16).

Output: Two fully connected layers to reduce the dimensions from 16 to the desired output size.

2. GRUModel_emotions



Architecture: Apart from a similar structure of GRU layers (256, 64, and 32), this model incorporates emotion data by first projecting emotion labels (one-hot encoded) into a 16-dimensional space and then adding this to the output of the GRU network.

Output: Uses a combination of emotional data and text embeddings to predict changes, fostering richer feature integration.

Dataset and Training

Both models were trained on the EFRDataset, prepared specifically for EFR:

Data Composition: Includes sentence embeddings, labels for emotional sequences, and ground truth for output labels indicating shifts in emotional states.

Training Procedure: Models were trained over 30 epochs, using loss functions and metrics suitable for classification tasks like cross-entropy loss and F1-Score.

Results

GRU-Val Weighted F1-Score: 0.30591673359002275
GRU_emotions Val Weighted F1-Score: 0.6880068166184801

Conclusion

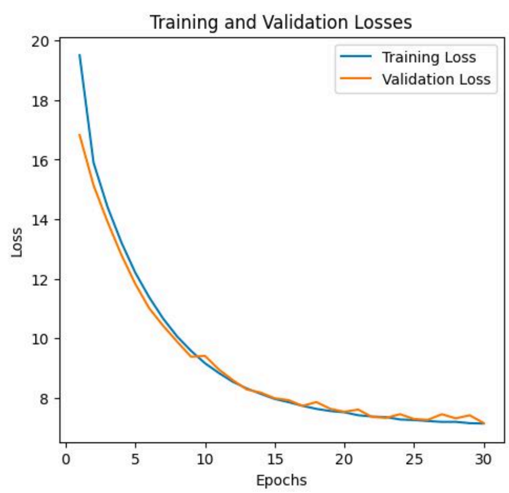
The emotion-augmented GRU model (GRUModel_emotions) demonstrated superior performance on both the training and validation sets compared to the basic GRU model. Specifically, it showed lower loss and significantly higher F1-Scores. This improvement is attributed to the integration of emotional context via the one-hot encoded projection which effectively bridges linguistic and emotional data, thereby enhancing the model's capacity to recognize nuanced emotional shifts.

Reason for model differences:

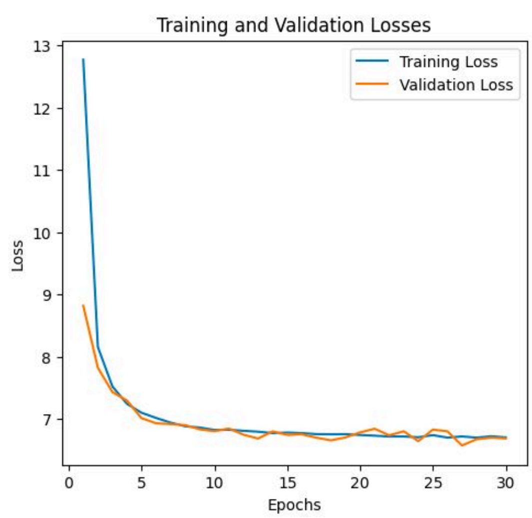
1. **Emotion Fusion:** Adding emotion-related information directly into the neural network's flow allows for a more holistic understanding of conversational dynamics.
2. **Architectural Efficiency:** The targeted use of GRU layers and emotion integration showcases an effective architecture for complex tasks such as EFR.

3. **Intuition:**The use of GRU (Gated Recurrent Unit) models in the context of EFR is crucial due to their ability to handle sequences, such as text or speech, which are inherent in conversational data.The additional emotional context in GRUModel_emotions helps the model recognize subtle cues in conversations that might indicate emotional shifts

Train and Val loss plots for M3 (GRU)-



Train and Val loss plots for M4 (GRU with emotions)-



Contribution

1. Sentence Embeddings - Nalish Jain(2021543)
2. M1 - Aniket Malik (2021231)
3. M2 - Sanmay Sood(2021095)
4. M3 - Shobhit Pandey(2021287)
5. M4 - Nalish Jain(2021543)
6. Inference - Sanmay Sood(2021095) and Nalish Jain(2021543)
7. Report - Aniket Malik (2021231) and Shobhit Pandey(2021287)

For test dataset during the demo we will have to calculate bert embeddings for the dataset on colab and then load the dictionary in the inference file