Design Credit Project

Topic : Speech to Text Conversion

PRESENTED BY

Bobbili Venkata Thrinath(B21CS017)

Project Mentor: Dr.Binod Kumar

Motivation

- **Speech to text** is applied to generate transcripts, captions or other written text that businesses today need.
- It works by “translating” speech into word-for-word written out formats. Every time you are using Siri or watching videos with captions, you're likely witnessing speech to text in action.
- Accessibility for the Disabled: A person with hearing loss can understand what is being said by using closed captions and speech recognition software to convert spoken words into text.

Dataset

- I used LJSpeech-1.1 Dataset
- The dataset contains 13,100 short audio clips of a single speaker reading passages from 7 non-fiction books.
- Each audio file has a corresponding text file containing the transcription of the spoken content.
- The transcriptions are provided in a metadata file named metadata.csv, where each row contains the file ID, the text transcription, and a normalized version of the text

PreProcessing the Data

- **VectorizeChar Class**: Converts text into a sequence of numerical indices.Defines a vocabulary including letters, some special characters, and positional markers.Maps each character to an index.Truncates and pads text to a fixed length, converting characters to their corresponding indices.

- **Audio Processing**: Reads .wav files and Computes the Short-Time Fourier Transform (STFT) to create spectrograms and then Normalizes the spectrograms, Pads them to a fixed length.

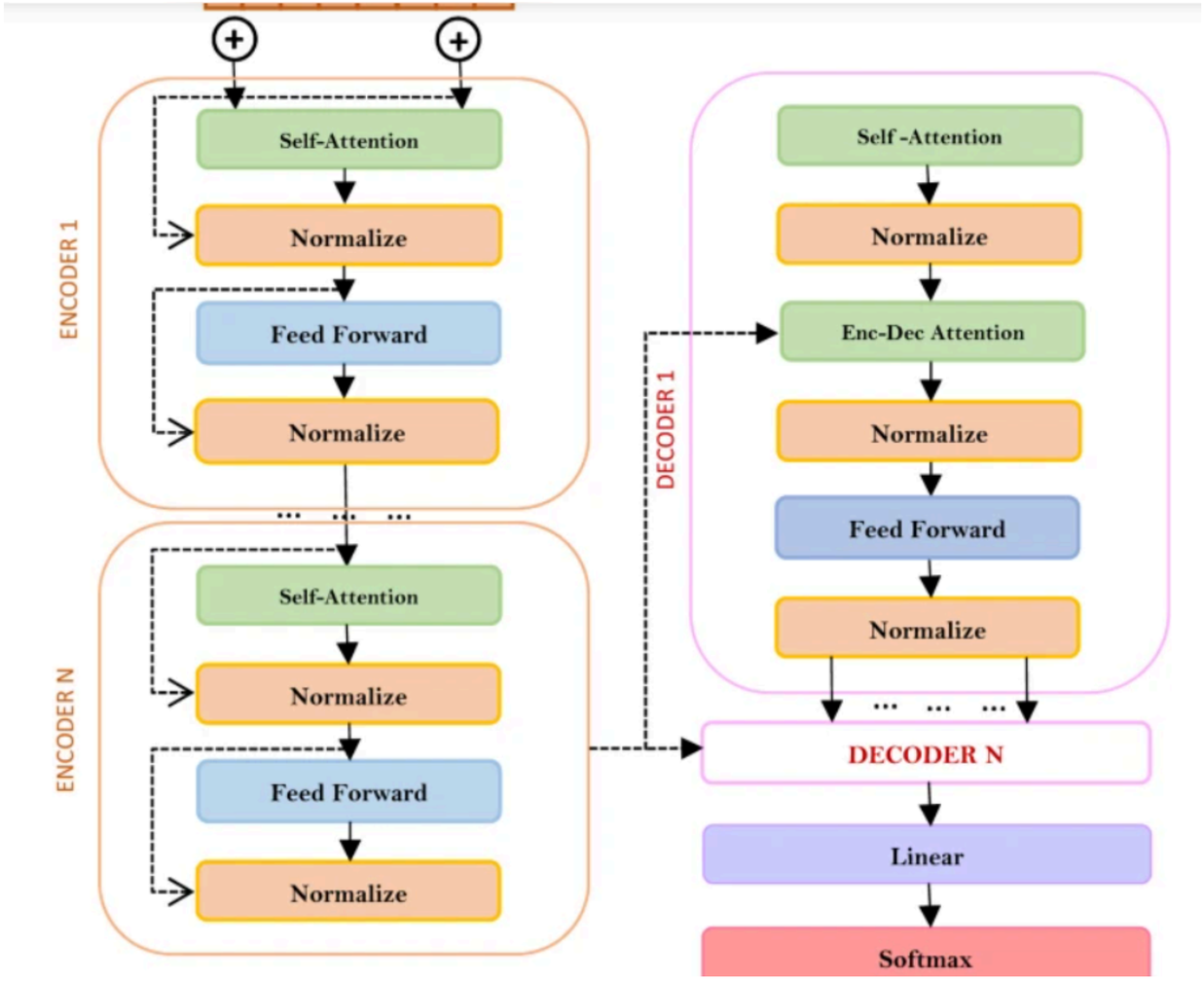
- **Combined Dataset Creation**: Create_tf_dataset() by combining audio and text datasets.Batches and prefetches the data for efficient training

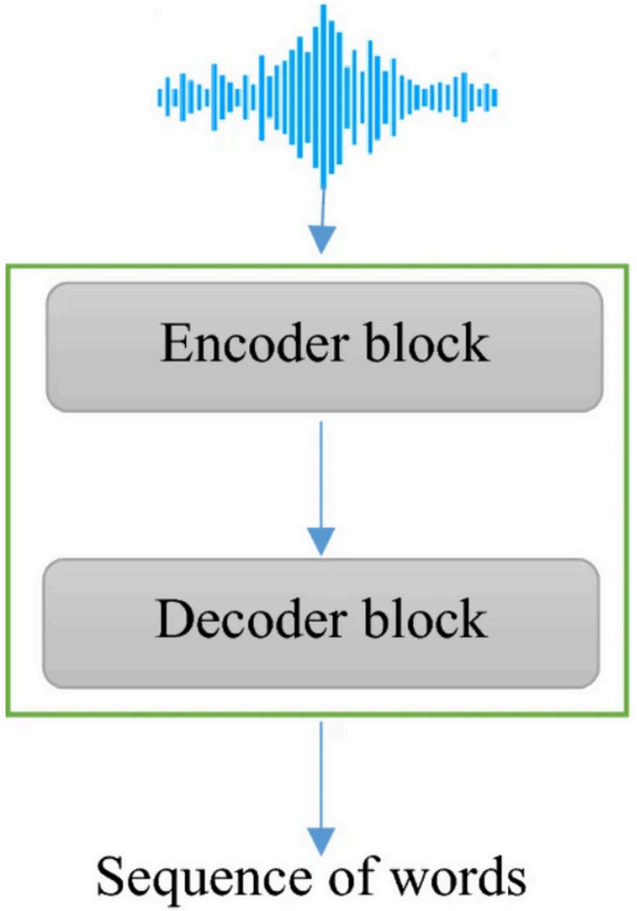
- **TokenEmbedding class** is designed to embed token indices and add positional encodings. It contains an embedding layer (self.emb) that converts token indices into dense vectors and another embedding layer (self.pos_emb) that provides positional encodings
- In the call method, it converts token indices to embeddings, generates positional encodings, and adds them to the token embeddings, which helps in providing sequence information to the model.

- **SpeechFeatureEmbedding class** processes audio features using a series of convolutional layers. It consists of three 1D convolutional layers (self.conv1, self.conv2, self.conv3), each with ReLU activation.
- In its call method, the input audio features are passed through these convolutional layers sequentially to extract higher-level representations from the raw audio input.

Transformer Architecture

- The Transformer model consists of one large block, which in turn consists of blocks of encoders and decoders .
- The encoder takes as input the feature vectors from the audio signal X = (x1,...,xT) and outputs a sequence of intermediate representations.
- Further, based on the received representations, the decoder reproduces the output sequence W = wm = (w1,...,wM)
- The encoder converts the vector of acoustic features into an alternative representation, and the decoder predicts a sequence of labels from the alternative information provided by the encoder, then attention highlights the significant parts of the frame for predicting the output


ENCODER 1
Self-Attention
Normalize
Feed Forward
Normalize
ENCODER N
Self-Attention
Normalize
Feed Forward
Normalize
DECODER 1
Self -Attention
Normalize
Enc-Dec Attention
Normalize
Feed Forward
Normalize
DECODER N
Linear
Softmax


Encoder block
Decoder block
Sequence of words

Implemented Custom Learning Rate Scheduling

- After adding a custom learning rate scheduler(**CustomSchedule**) there is a significant impact on the Prediction of output accuracy
- Initially, the learning rate starts low and gradually increases. This helps stabilize training early on by avoiding large updates that can destabilize the model.After the warmup, the learning rate decreases gradually. This allows the model to converge more smoothly by making smaller updates as it approaches the optimal solution.
- Additionally **DisplayOutputs:** A callback for visualizing the model's performance at intervals (every 10 epochs). It compares model predictions with actual targets to monitor progress.

Results

- At epoch 1

```
target:     <the seemingly most courageous was selected to lead the way.>
prediction: <the o the t the athe s the te o liofin the the the there athate as te te o wand an aro r the the t tison
as the the the ate then te athe there that on the the the te onisere tere ane o th hen e t the

target:     <when he was finally apprehended in the texas theatre. although it is not fully corroborated by others wh
o were present,>
prediction: <the o the t the athe s the te o liofin the the the there athate as te te o wand an aro r the the t tison
as the the the ate then te athe there that on the the the te onisere tere ane te a hen e t the
```

- As we can observe from above since we have just started the training of the model there is large difference between actual speech text and predicted text

- At epoch 31

```
Epoch 31/51
203/203 ━━━━━━━━━━━━━━━━━━━━ 0s 342ms/step - loss: 0.4699target:     <the seemingly most courageous was selected to l
ead the way.>
prediction: <the semingly most corages was selected to the way.>

target:     <when he was finally apprehended in the texas theatre. although it is not fully corroborated by others wh
o were present,>
prediction: <when he was finally coroberated by others who were present fully coroberated by others were present.>

target:     <courvoisier, when put on his trial, pleaded not guilty#>
prediction: <could on his trial, pleaded not guilty.>

target:     <again the following year the inspectors repeat their charge.>
prediction: <again, the following year the inspectors repeat the inspectors repeat the inspectorsh,>
```

- At epoch 51

```
Epoch 51/51
203/203 ━━━━━━━━━━━━━━━━━━━━ 0s 342ms/step - loss: 0.3852target:     <the seemingly most courageous was selected to l
ead the way.>
prediction: <the semingly most crragious, was selected to leaving le be the way.>

target:     <when he was finally apprehended in the texas theatre. although it is not fully corroborated by others wh
o were present,>
prediction: <when he was finalling apprehended in the texast the atred by others# were present,>

target:     <courvoisier, when put on his trial, pleaded not guilty#>
prediction: <courvisier, puth on his trial fleaded not guil, pleaded not guilty.>

target:     <again the following year the inspectors repeat their charge.>
prediction: <again, the following year the inspectors repeat the following year the inspectorsh.>
```

- We can observe that from epoch 1 to 51 the loss is decreasing and we can also clearly visualise that the actual speech text and predicted text are having major similarities

- With each epoch, the model becomes better at mapping inputs to correct outputs, reducing discrepancies between predictions and actual values.

Thank you