Centre for Innovation (CFI)
Indian Institute of Technology, Madras
Application for the Post of Project Member
AI RAHMAN **AI Club (2024-2025)**

Name: DIPANSHU PANDA		Nickname: DEV
Roll.No: CE22B004	Room: 3029	Hostel: CAUVERY
Phone: 7978959132	CGPA: 9.74	Email: dipanshu.panda75@gmail.com

Section A: Managerial Questionnaire

1. State your motivation to become an AI club project member. What do you feel makes you a decent fit for this job? Justify by stating your skills/strengths and previous experience (if any). (Explain why you're choosing AI/ML not just for a PoR but out of interest.)

My primary motivation for joining the AI Club as a Project Member is to explore the world of Artificial Intelligence and, most importantly, gain more experience in developing projects.

- Artificial Intelligence as a field has always piqued my interest but it was after witnessing the amazing projects on display by the Analytics Club and the CVI club at the CFI Open House that I realized the beauty of this field.
- The scope of Artificial Intelligence is vast and is only going to increase with time and hence, as an enthusiast of the same, my motivation is to contribute to this cause to the best of my abilities and also increase my knowledge along the way. So, what better place to do this, than the AI Club?

I feel I am a decent fit for this job because of the following skills/strengths that I possess-:

- **Expertise in NLP-:** I am currently enrolled in the course on Natural Language Processing (CS6370) offered by the CS department of IIT Madras. With this course, I have had the experience of reading several research papers based on the various building block problems of NLP and most importantly, the techniques used to solve them like generative models, sequence models, Bayesian inferencing, etc. which are of course not, domain-specific. In the app itself, there is a dedicated section for sequence models and applications in NLP and hence, I think I can be of great help. I have also earned certifications in the Machine Learning Specialisation offered by Coursera(Andrew Ng).
- **Experience of working in teams and Research Projects-:** As a part of the NLP course project, I am currently writing a Research Paper with two of my teammates on the Techniques used for Information Retrieval Systems like Latent Semantic Indexing, Case Retrieval Nets, Latent Dirichlet Allocation and a comparative study of these models. As mentioned in one of your sessions, I am your guy if you want someone to focus on research more than glittery tricks to just get things done.
- **Curiosity-:** This is one of the main reasons why I am applying in the first place. I am very passionate about exploring Artificial Intelligence and working with like-minded people at the AI Club. This is one of my strengths as it helps me sustain my interest in the project at hand and not give up midway.

- **Dedication-:** As an individual, I can assure you that I will give my absolute best in completing the project with utmost dedication. I will make sure that I perform the tasks given by the Project Leads to the best of my abilities and strive to improve.
- **Programming skills in Python-:** I am well-versed in the Python programming language and have implemented various models using Tensorflow, Scikit, NumPy, etc. As mentioned above, I have extensively used the programming language in the courses and projects.

2. Commitments/PoRs:

a. How much time do you think you can commit to AI Club weekly?

The AI Club will be one of my topmost priorities. I am confident that I will be able to juggle between this PoR and other obligations. This is something that I am passionate about and hence, I will pursue this dedicatedly and actively without making excuses. Roughly, I will be able to devote about 2/2.5 hours a day on the weekdays and 3.5 hours on the weekends which takes the total to 18 hours in a week. Again, this is just a rough estimate and not a hard deadline. I am happy to devote more time to the project if required of me.

b. What other PoRs/activities do you plan to take up next year? In case of clashes, how will you prioritize your role as a project member?

I am not interested in any other PoR/activities to take up in the next year. This is the only PoR I am interested in. I will always make sure that as a project member, I contribute to the team actively despite any other obligations like academics or other projects. And rest assured, I will not leave the project midway under any circumstance. As I have mentioned earlier, I am not treating this as just a PoR but as a great learning opportunity and an experience that would help me grow.

Section C: Project-Specific Questionnaire-:

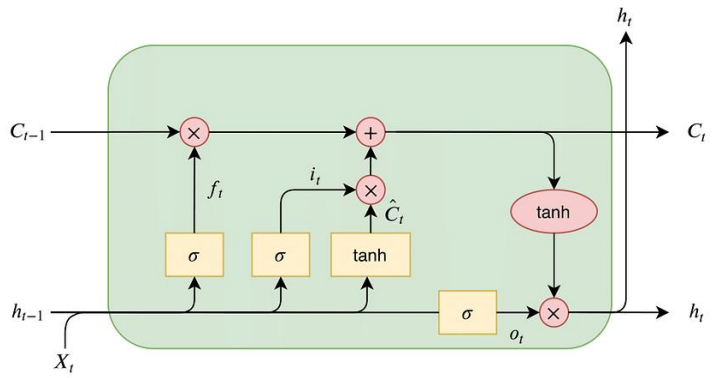
Sequence 2 Sequence Models

a.How do RNNs differ from traditional ANNs in capturing long term dependencies? Why do you think this is important for music generation?

Recurrent Neural Network is a type of neural network where the output from the previous time step is fed as input input in the current step. This is extremely crucial for tasks dealing with sequential data like sentiment analysis and machine translation in NLP, and time series forecasting.

- This is because in traditional neural networks, all inputs and outputs are independent of each other but in cases such as predicting the next word of a sentence, the knowledge of previous words is required and hence, the need to ‘remember’ previous words. RNN achieves this through the Hidden state. Also referred to as Memory state, it remembers the previous input to the network. So, for example, the prediction of an ANN remains same for a test case, but for an RNN, it might change depending upon the previous inputs to the RNN.

- In the case of sequential data, we need the neural network to be flexible with respect to the length of sequential data required to make a prediction. Traditional ANNs are very rigid in this matter, whereas RNNs can accommodate varying sequence lengths.
- RNNs have the same input and output architecture as any other deep neural architecture. However, differences arise in the way information flows from input to output. Unlike Deep neural networks where there are different weight matrices for each dense network, in RNN, the weight across the network remains the same. It calculates the hidden state H_i for every input X_i. This reduces the complexity of parameters unlike Artificial Neural Networks.
- The fundamental problem with RNNs is that they are harder to train for long sequences due to the vanishing/exploding gradient problem. Due to recurrent loop of the RNN, the weight of the loop grows exponentially as the number of sequential data points increases. As a result, if the weight is greater than 1, it results in an exploding gradient and if it is less than 1, we have a vanishing gradient.
- LSTM architecture is used to overcome this problem. It follows the principle of selective read, write, forget. The LSTM has two states- Cell state which has the long term memory and the Hidden state having the short term memory. There are 3 gates- Forget gate determines the percentage of long term memory to be remembered, input gate determines the potential long term memory to remember and the output gate determines the potential short term memory to be remembered.
- LSTM Architecture-: I have implemented this for the Sentiment Analysis

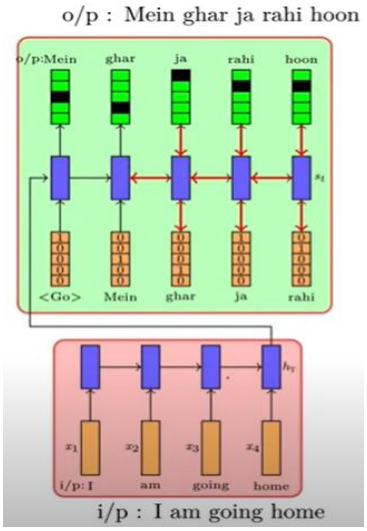


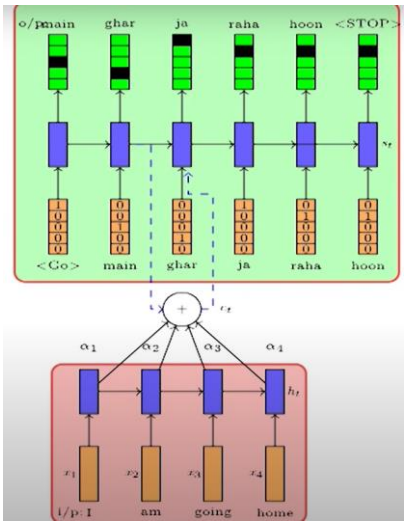
Of course, in the context of music generation, RNNs are very important because music is also a form of sequential data.

- RNNs create output based on a context of history, which is very useful for musical compositions since they, rely on a global context and theme of the entire piece. Otherwise, imagine writing notes to a song and forgetting what you have already written, the song would not have a theme.
- There are already existing LSTM networks that generate music such as Skuldur's Classical Piano Composer in which the underlying concept is of the LSTM in order to ensure that the generated piece has a coherent theme.

b. Go through the reference links given below and provide a summary of encoder, decoder blocks . Give an intuitive explanation on attention mechanism.

- Encoder-decoder architecture is a fundamental framework used in various fields and for solving a wide variety of problems. An encoder block is essentially used to process the input data and transform it into a different representation and pass it onto the decoder block. The decoder block then processes this representation and decodes it to produce the output.
- At its core, encoder-decoder architecture operates on the principles of feature extraction and data transformation. The encoder network processes the input data, extracting essential features and creating a concise representation. This representation is then decoded by the subsequent network to generate meaningful outputs. Of course, this might sound very general and hence, to drive home my point, I would like to use an example of Machine Translation.
- Let us say, I want to translate a sentence from English to Hindi, my input is a sequence of words and so, is my output. Hence, I would use an RNN(maybe also LSTM) as an encoder block. It would take in the embeddings of the sequence of words in English(maybe Word2Vec or one-hot) and give me a representation carrying the knowledge of the entire sentence(last time step of the encoder).
- I would then pass the last time step of the encoder as the s_0 (state at the 0^{th} time step) to another RNN, my decoder block, that would generate the sequence of embeddings of the words in Hindi. At any time step, the decoder block essentially, has the knowledge of the original English sentence (s0) and also the knowledge of the Hindi words that have already been generated at previous time steps. Now, the translated Hindi sentence is constructed. Here, is a picture highlighting the process in detail-:





Attention Mechanism-:

- But, there is a major limitation in this model because, this is not how humans translate. The encoder essentially, reads the entire sentence and encodes it once and for all. The decoder is then overloaded with the knowledge of the entire sentence.
- But, is the entire sentence important at every timestep? Certainly not. Humans try to produce each word in the output by focusing only on certain words in the input. So, essentially at each timestep, we come up with a probability distribution over all the input words which tells us which words we need to pay attention to. This is the intuition behind integrating the Attention Mechanism in a vanilla

encoder decoder system.

- For instance, in the above example, for translating to “Main”, we just need to pay attention to “I”. Similarly, for “ghar”, only pay attention to “home”. But for “jaa rahi hoon” we need to pay attention to both “am going”. The model needs to learn to pay attention to certain parts of the sentence at a timestep.
- For integrating this attention mechanism into the model, we define attention weights or probabilities to all the words in the input sentence at each timestep. The weighted average of the encodings is then passed onto the decoder block at every timestep rather than overloading the decoder at the 0^{th} timestep.
- Of course, these attention weights also become parameters of the model that have to be learnt as we train the model on different sentences. But that can wait for now. The second picture illustrates what I just explained.

c. (Brownie Question) Nowadays, Graphics Processing Units have taken over the world of Machine Learning. With their extreme parallelization capabilities, they have revolutionized computation, reduced training time by a significant amount. As seen in the above sequence-to-sequence models, there is not much parallelization involved. Each word is processed only after the previous word, the computations are mostly sequential. Can you think of any way to leverage the power of GPUs here (i.e. make computations parallel)?

Here, are some ways in which we can leverage the power of GPUs in sequence-to-sequence models-:

- Batch Processing: A batch of sequences can be processed in parallel rather than processing a single sequence at a time. This makes use of the GPU's capacity to process huge batches of data effectively through matrix operations. Several sequences can be processed at once, which cuts down on the total training time
- Attention Mechanism: As seen above, sequence-to-sequence models employ attention mechanisms, where the model learns to focus on relevant parts of the input sequence while generating the output. The attention weight computations can be parallelized across the input sequence, allowing the model to compute the attention weights for all input features simultaneously.
- Beam Search Decoding: In sequence-to-sequence models, the output is usually produced one word at a time during the decoding stage. Instead, beam search decoding can be used, in which the model simultaneously investigates several candidate output sequences and chooses the most promising one. The decoding process can be parallelized, which could speed up the generation of the final output.
- Transformers: The Transformer architecture, which has become popular in recent years, is designed for this very purpose. In Transformers, the sequential nature of recurrent models (such as RNNs and LSTMs) is replaced with a fully-connected, self-attention-based approach. This allows for more parallel computations, as the model can process all input features simultaneously and compute the necessary attention weights in parallel.

Digital Audio

a. What is sampling of signals? Mention the frequencies at which a music audio is sampled and why. Using the concept of bit depth mention the relationship between audio quality, dynamic range and sampling frequency.

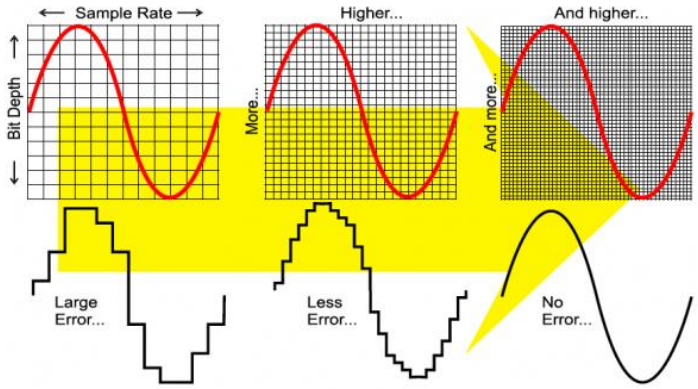
Sampling of signals is the process of creating discrete digital signals from the samples of a continuous analog signal. In the context of audio, sampling involves capturing snapshots of a waveform at specific points in time.

- A music audio is typically sampled at the frequencies of 44.1 and 48 kHz, also known as sampling rate. This is based on the Nyquist Theorem, which states that to accurately recreate a signal, sampling frequency must be at least twice of the highest frequency present in the signal.
- Since, the human hearing range itself is 20Hz to 20kHz, a sampling frequency of 40kHz can be used but we need, a very powerful low-pass filter to prevent audible aliasing. By placing the Nyquist frequency outside of our hearing range, we can use more moderate filters to eliminate aliasing without much audible effect.
- Hence, 44.1kHz is used for audio CDs while 48 kHz is used in audio for video like movies. The higher sampling rate leads to more measurements per second a closer recreation of original audio.

The concept of audio bit depth determines the number of possible amplitude values we can record for each audio sample. It refers to the number of bits used to represent a sample and hence, the precision or resolution of the digital representation is determined by bit depth.

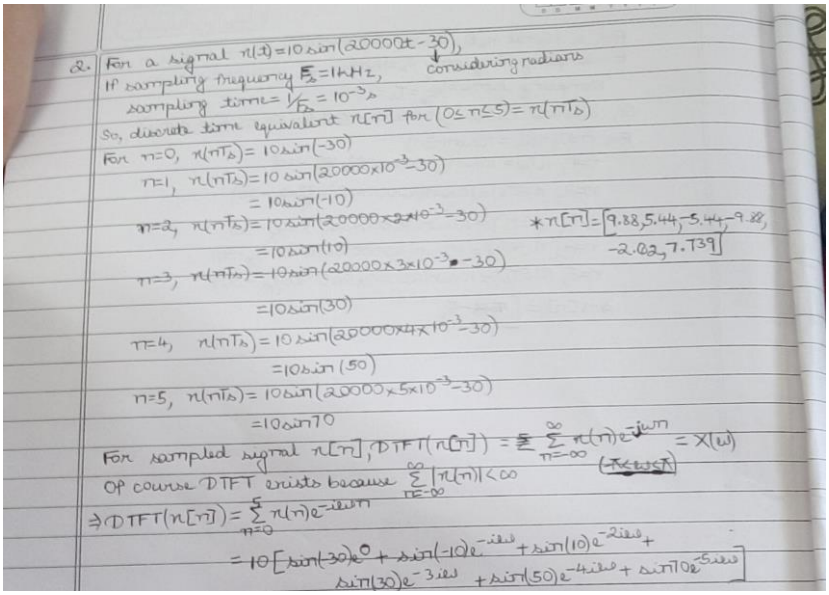
Here is how audio quality, sampling frequency and dynamic range are related-:

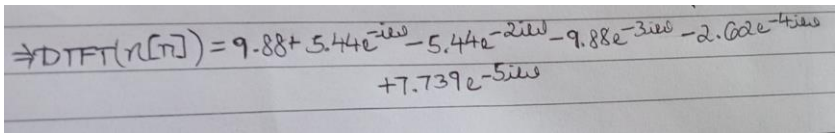
- A higher bit depth along with a high sampling rate increases the audio quality of the reproduced signal. This is simply because, increasing the audio bit depth, along with increasing the audio sample rate, creates more total points to reconstruct the analog wave.
- We have more amplitude values available to record. As a result, the continuous analog wave's exact amplitude is closer to an available value when we sample. So, the digital approximation becomes closer to the original analog wave.



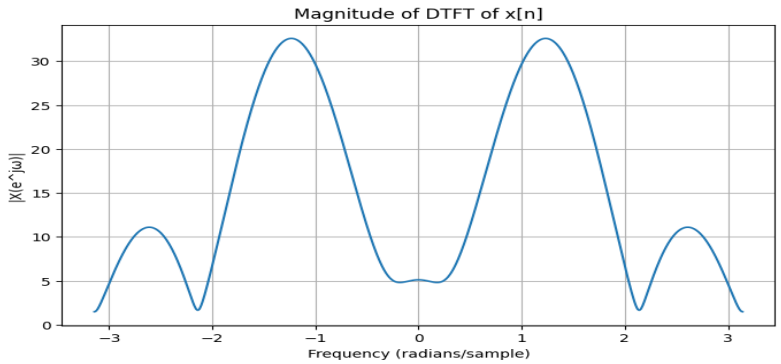
- Higher bit depths allow for a greater dynamic range, which is the range between the loudest and the quietest sounds that can be accurately represented. Dynamic range is the difference between the low volume and high volume sections of a recording(measured in decibels).

b.The Fourier transform is a mathematical formula that transforms a signal sampled in time or space to the same signal sampled in temporal or spatial frequency. In signal processing, the Fourier transform can reveal important characteristics of a signal, namely, its frequency components. In mathematics, the discrete-time Fourier transform (DTFT) is a form of Fourier analysis that is applicable to a sequence of discrete values. The DTFT is often used to analyze samples of a continuous function. The term discrete-time refers to the fact that the transform operates on discrete data, often samples whose interval has units of time. Consider a signal x(t)=10 sin(20000t − 30). Sample the signal at a sampling frequency of 1kHz and obtain the discrete time equivalent x[n] (for $0 \leq n \leq 5$) ,where x[n]=x(nTs) [Ts is the sampling time] . Calculate the discrete time fourier transform for the sampled signal x[n].





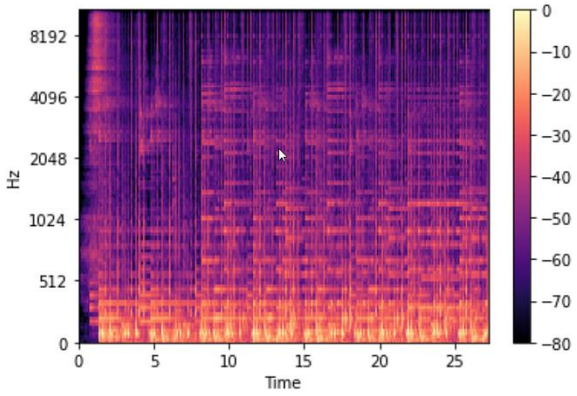
A discrete time signal is represented in the frequency domain using DTFT. This DTFT represents the frequency content of the discrete sequence x(n). Therefore, by computing the fourier transform of the discrete time sequence, the sequence is decomposed into its frequency components. For this reason, DTFT is also called the signal spectrum. Below is the plot for various values of omega in the range of (-pi to pi)



c.Write a shortnote on how fourier transforms can be used to analyze music audio signals. (hint: check Short Time Fourier Transforms and Mel Spectrograms).

The most challenging aspect of working on an ML project like this with audio data or time-dependent signals is that it is impossible to work with raw signals. We have to preprocess these signals and extract useful features before getting into the nitty-gritty. Sound is simply defined as a pressure wave created by a vibrating object. Visualizing sounds, hence, can be more aptly defined as visualizing certain sequences of vibrations with varying air pressures with respect to time. So, how to do this preprocessing and extract useful features from audio signals:

- As mentioned before, any signal is a sequence of variations in a specific quantity with respect to time. For audio signals, this quantity is air pressure. We can use Python libraries like Librosa to capture this digitally. Now, to extract useful information from this fundamental waveform, we apply the fourier transforms.
- Fourier transforms are mathematical constructs that take in a signal from the time domain and decompose it into its individual frequencies, resulting in a frequency domain(spectrum). We can apply fast fourier transforms on short-time windowed segments of the audio signal and then overlap all of these segments.
- This is achieved by the Short-time Fourier Transform(STFT). STFT separates the signal into time windows and applies FFT on each of these windows. This gives us the visualization of the signal's frequencies as it varies over time.
- But for analysing music audio signals, the human perception of sound is necessary to consider. The human perception of sound is concentrated in very small frequency and amplitude ranges. Hence, the frequency scale is converted to a log scale and the loudness(decibels) is taken as a color scale. The resulting output is the Spectrogram.
- Humans also perceive frequencies in a non-linear manner. The "mel scale",originating from melody, regularizes the frequency intervals between any two given notes. To convert f hertz to m mels we have many empirical formulas-:
- After this regularization, and converting hertz to the mel scale in the spectrogram, we finally get the Mel-Spectrogram. Generating the Mel-Spectrogram is the most fundamental unit of audio processing. After this is done, there are multiple use cases where the Mel-Spectrogram plays a significant role.



d. What is MIDI representation in music? Mention its advantages over digital audio. What are limitations of MIDI representations?

MIDI and digital audio are different ways of recording information about sound. MIDI is a set of instructions about generating a sound (using a MIDI device), whereas digital audio represents an actual sound wave. Talking about digital audio, binary data is used to represent sound. It entails transforming the analog sound waves into a digital version that is stored, handled and transferred using digital technology.

MIDI, on the other hand-:

- It is the abbreviation for Musical Instrument Digital Interface. It is a protocol that enables electronic musical instruments, computers and other devices to communicate with one another.
- So, instead of recording the actual sound, it is a method of encoding musical notes and performance data as digital information. It's only a set of instructions or commands about how to produce a sound rather than an actual representation of the sound.
- Just like we use embeddings to encode not just the identity of a token but its relationships with other tokens in Natural Language Processing and then use these encodings for tasks like sentiment analysis, machine translation, etc. MIDI allows us to sequence music. This means that one can string together a series of notes and chords. just the instructions, of course, not the actual sounds, to form a song or piece of music.

I will be listing down the advantages and limitations of MIDI representations with the help of a comparison with digital audio-:

Advantages-:

- MIDI files are much smaller and compact as compared to digital audio files. Digital audio can be 20 times larger than MIDI files. This makes storage and transmission of digital MIDI files much more easier.
- MIDI files are easy to edit. Since they contain instructions, and not actual representations of sound waves, it's very easy to change the sequence or type of instructions in a MIDI file. This is a lot harder to do with digital audio. With digital audio, you would need to alter the very sound that the data represents and there's no easy way of doing this other than by re-recording the chord.
- MIDI files are portable across instruments. Since MIDI files contain only instructions, the actual instruments that we choose to play the instructions with can be selected and changed at will (provided,

of course, that the instruments are MIDI-enabled). With digital audio, once an instrument is recorded it can't be changed.

Limitations-:

- MIDI data only works with MIDI-enabled instruments. We can't use MIDI data to capture voice or natural sounds, it can only be used to send instructions to MIDI-enabled devices that produce their own sound. With digital audio, just about any sound or any instrument can be captured, either directly through line-in connections or through microphones connected to a suitable audio interface.
- MIDI is focused more on music production. Talking about its history, MIDI was developed by the music equipment industry and was first deployed at scale for use in synthesizers. MIDI has, to date, had limited use outside of core music production.

Either way, it is interesting to see that MIDI actually depends on digital audio. Because when a MIDI file is sent to a MIDI device, the sounds generated are from sampled digital audio files. The MIDI file merely instructs or "triggers" the audio file to play.

Coding Task- Notebook Links:

Technical Questionnaire Quantile Regression:

https://colab.research.google.com/drive/1Yp3BWyxzBGL4guBv_49eaSRrHwEOti6R?usp=sharing

Project Specific RNN:

https://colab.research.google.com/drive/1uiOn6Flttpw8XON_e8ujD_SCCPHDvjPn?usp=sharing

THANK YOU FOR GOING THROUGH MY APP