AI Club Project Member Application

Harshith M R

May 2, 2024

1 Managerial Questionnaire

1.1 Essentials:

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 motivation to become project member comes from my passion towards AI and ML. I have been interested in AI when I saw Iron man interacting Jarvis which was so cool!. I was amazed by the fact that a software can do tasks which only humans can. I would be a decent for the job as I have previous experiences in this field. I had participated in Inter IIT 12.0 and had won bronze in it as a team. I also took part in Convolve 2.0 in which 1600 teams took part and our team was placed 3rd. I also keep with recent research papers and try to implement them. I had implemented papers like food vision and skimlit.

1.2 Commitments/PoRs:

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

I would try my best to complete the work that is assigned to me and help my fellow coordinators instead of keeping track of the time I spend.

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 will apply for AI contingent member for this academic year apart from this. The peak of an AI project member happens during the open house and the research conclave. The peak of a contingent member happens during Inter IIT TechMeet. Usually research conclave takes place in the first week of November and it is the season for the release of Inter IIT problem statements. I will complete the work for research conclave much before so that I will be able to work on the Inter IIT problem statement. Open House will be after Inter IIT in March so that wont cause any problem.

2 Technical Questionnaire

2.1 Common Technical Questionnaire:

Link for the notebook: Quantile Regression

Bonus Question: Play around with the value of τ to find what value achieves convergence quicker.
I wrote a code to test values of τ from 0.1 to 0.9 with step size as 0.1. This graph gives the number of epochs it took for each τ value to get predicted value closer to the actual value. Since we only have two samples here the loss will be lower for the 0.5th quantile since it is at the center between the samples and increases in both the ways.

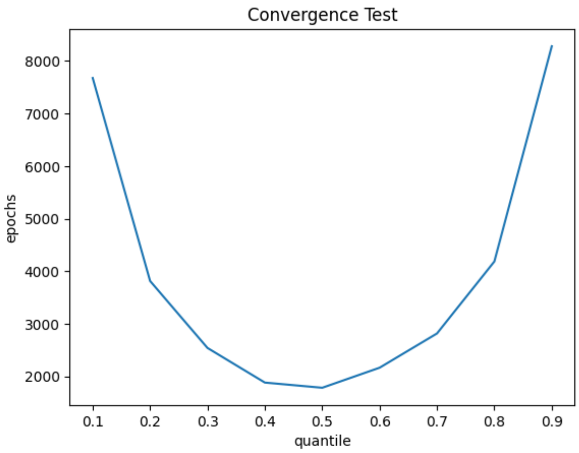


Figure 1: Epochs vs Quantile Value

2.2 Project Specific Questionnaire

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 by which continuous time signals are converted into discrete time signals for the ease of storing it as continuous signals consist of infinite points and storing it would require infinite memory.

Music audio is typically sampled at 44.1 KHz because humans can hear sound up to a frequency of 20 KHz and according Nyquist Shannon Sampling Theorem in order to accurately reproduce a signal it should be sampled at a frequency twice its highest frequency. The extra 4.1 Hz from twice of 20 KHz is because of the limitations in analog to digital converter and this 44.1 KHz was standardized by sony.

Bit depth is the number of bits used to represent the amplitude of each sample in digital audio. Higher the bit depth, better the resolution and greater the dynamic range. Dynamic Range is the difference between quietest and the loudest sounds that can be well represented. So in conclusion greater the bit depth lesser loss in conversion from analog to digital audio. Any sampling frequency greater than twice the Nyquist frequency with some buffer is fine. If we sample 2 samples per cycle it is enough to accurately construct the wave back without losses according to Nyquist theorem.

b. I have attached the solution handout in the back of latex file.

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

The Short-Time Fourier Transform (STFT) is a signal processing technique used to analyze non-stationary signals, such as music audio signals, by providing information about their time-varying frequency content. The STFT is different from the normal Fourier Transform as the FT just decomposes the signal into constituent frequencies but does not give info about how it varies over time. STFT works on the principle of a window moving over the signal and the FT is computed for each window.

This is useful for analysing music audio signals as it captures time varying nature of music signals.

- To track Pitch and Melody
- Timbre Analysis
- To efficiently store audio signals
- Source separation

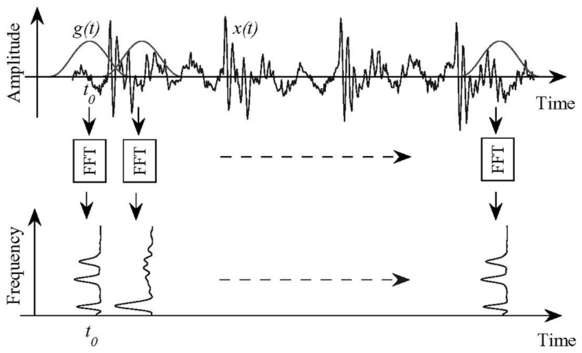


Figure 2: Short Term Fourier Transform

Mel Spectograms are useful for analysing music audio signals because they capture relevant features like humans. Mel spectograms are basically STFT incorporating additional features like converting the frequency scale to mel frequency which is similar to how human perceive sound. The use of this is this can be fed into ML models as training data to make the model understand music.

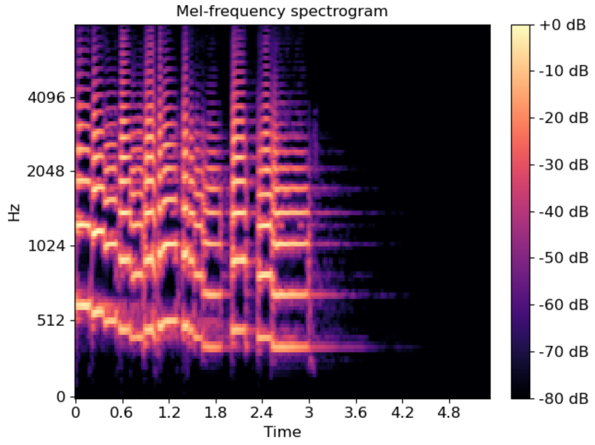


Figure 3: Mel Spectogram

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

MIDI (Musical Instrument Digital Interface) representation in music does not capture audio but it captures musical notes, timings and pitch information.

Advantages of MIDI:

- Compact file size
- Edit/Add notes after recording in MIDI in case you made a mistake while recording
- Cross platform portability (MIDI data can be played in different hardware/software devices)

Limitations of MIDI:

- Since MIDI only represents the notes played by musical instruments the audio can only be replicated identically if the instrument is the same as used in production.
- MIDI data cannot be used to emulate voice, timbre, tone, or expression of the sound.
- Higher Cost.

MIDI has it own advantages and limitations compared to digital audio.

Figure 4: MIDI

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?

Traditional ANNs differ from RNNs as they can't use the reasoning from the previous events to get insights for the current event whereas RNNs have feedback loops which pass the information of the previous event to the current event which is the major advantage which RNNs have over ANNs.

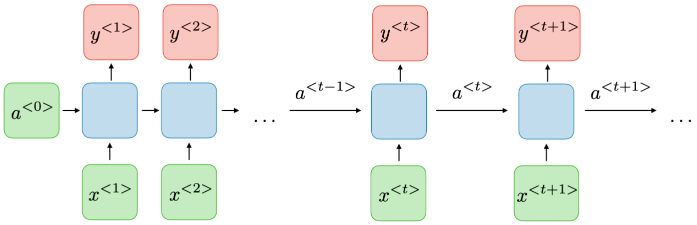


Figure 5: RNN

Even though RNNs have these feedback loops, the basic RNNs are not so often used in practice for exploding/vanishing gradient problems. As the weight in the feedback loop gets multiplied subsequently from one unit to another this makes the output too small or long. This result creeps into the gradient calculation which makes simple RNNs not so efficient.

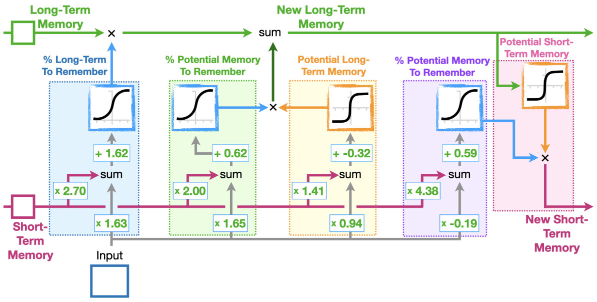


Figure 6: LSTM

Due to counter this gradient problem LSTM networks were introduced which have two types of memories long term and short term. LSTM network consists of 5 parts: Forget gate, Percentage of long term memory to remember, Potential long term memory, Percentage of short term memory to remember and Potential short term memory. It also consists of sigmoid and tanh activation functions within its architecture.

Music generation tasks are sequence problems just like text or image generation where the next note or sound that is to be generated depends on the previous sound. If the information about the previous generated music is not fed into the network the thus generated music will be noise and wont rhyme with the previous. With the help of LSTMs and RNNs we will be able to feed the features of previously generated music to the current to generate much more pleasant and nice to hear sounds.

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

The encoder and the decoder block consists of a set of LSTM layers. Each layer of LSTM consists of LSTM cells. Each word passed is passed to the first layer of LSTM in encoder block which outputs long term and short term memories to the next LSTM cell and the next word is also passed to it and this continues to get the final context vector which is ready to be passed to the decoder block. The first word that is passed on to the decoder block is the EOS end of sentence token with context vector. The output of the short term memories is passed onto the feed forward layer with softmax activation to get the next predicted word. The next predicted word is then passed to the subsequent cell along with the memories from the previous to predict the subsequent words. The prediction of the decoder won't stop until it predicts EOS. While training even if the Decoder fails to predict EOS when it should, we terminate the process through a process called Teacher Forcing.

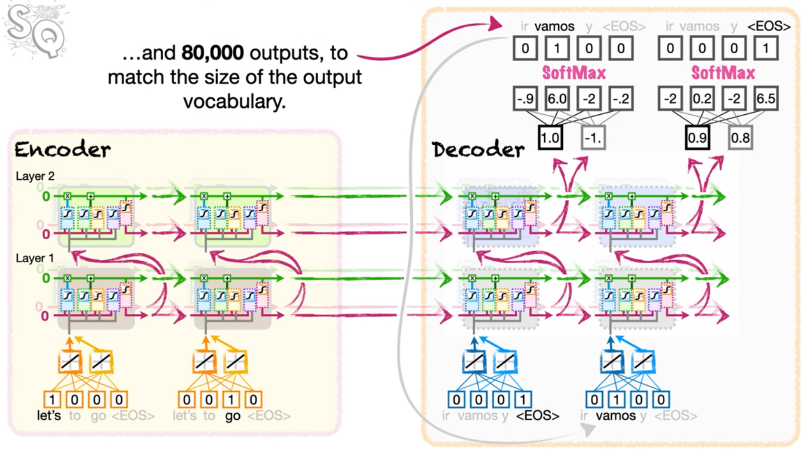
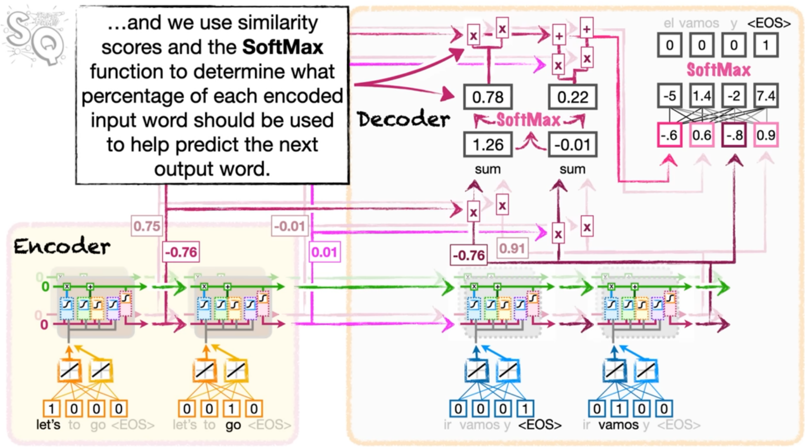


Figure 7: LSTM Encoder Decoder without Attention

Even though we use LSTMs as our backbone for encoder and decoder blocks, when we write long sentences the info about the initial words is forgotten. Thus attention was introduced. Instead of just connecting the final context vector of the encoder to the decoder block we also connect the outputs of each short term memory of the encoder block to each decoder block to determine which word of the input has the most influence in predicting the next word. The short term memory of each encoder is passed with the short term memory of the current decoder to calculate the similarity scores (often cosine similarity). These similarity scores with each memory are then passed to a feed forward layer with softmax to get the percentage relations of the decoder output with each encoder output. The probabilities are in turn multiplied with short term memories of the encoder to get the attention values. These attention values concatenated with decoder short term memory are passed on to another feed forward layer with softmax to predict the next word and the process continues till the decoder outputs EOS.



...and we use similarity scores and the **SoftMax** function to determine what percentage of each encoded input word should be used to help predict the next output word.

Figure 8: LSTM Encoder Decoder with Attention

The intuitive explanation for attention is that it focuses on relevant parts of the input which gives more context in predicting the output rather than the whole input. This attention mechanism mimics human-like behavior while reading a story. The key parts like the twists and the plot are much more relevant for predicting how the story goes on than some irrelevant details like wardrobe choice of the character.

c. (Brownie Question) Nowadays, Graphics Processing Units have taken over the world of Machine Learning. With their extreme parallelization capabilities, they have revolutionized computation, reducing 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)?

Transformer architectures were developed for extreme parallelization. The computation of Query, Key and Values for attention can be carried out independently without the need of the previous outputs. These computations can use GPUs for extreme parallelizability.

2.3 Coding Questionnaire:

Link for the notebook: Sentiment Analysis

DTFT.

$x(t) = 10 \sin(20000t - 30)$

time period $= \frac{1}{1000} = 10^{-3}$ sec.

$x[0] = 10 \sin(20000(0) - 30) = 10 \sin(-30) = -5$

$x[1] = 10 \sin(20000(10^{-3} - 30) = 10 \sin(-10) = -1.7364$

$x[2] = 10 \sin(20000(2 \times 10^{-3}) - 30) = 10 \sin(40 - 30)$

$= 10 \sin(10) = 1.7364$

$x[3] = 10 \sin(20000(3 \times 10^{-3}) - 30) = 10 \sin(30) = 5$

$x[4] = 10 \sin(20000(4 \times 10^{-3}) - 30) = 10 \sin(50) = 7.6604$

$x[5] = 10 \sin(20000(5 \times 10^{-3}) - 30) = 10 \sin(70) = 9.3969.$

$$x(e^{jw}) = \sum_{n=0}^{5} x[n]\, e^{\wedge}(-jwn)$$

$$X(e^{jw})$$

$$= -5(e^{0}) + (-1.7364) e^{-jw} + (1.7364) e^{-2jw} + (5) e^{-3jw} + (7.6604) e^{-4jw} + (9.3969) e^{-5jw}.$$