Generating Music
with Machine Learning

Team - Closed AI

Advait Shrivastava, Pranshu Nema and Samyak Jain

(2022201069,2022202029,2022201048)

Problem Statement

- Exploring the various ways in which a computer can be taught to generate music.
- For individuals looking for creativity and innovation in their music writing, learning algorithms will be a huge help.

Scope Discussed

- Experimentation on the presented methods.
- Try attempting something other than paper.

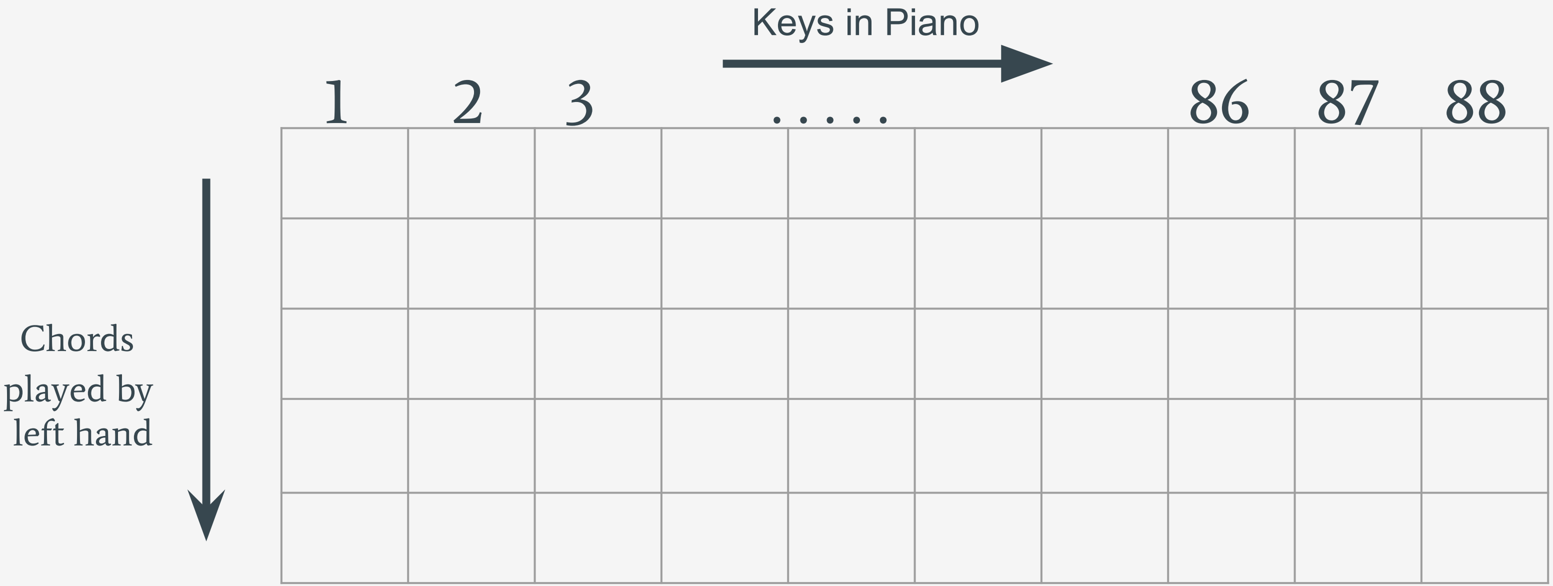
Methods Presented

Naive Bayes

- Purpose of this model is to make a distribution for which **keys** are pressed for a given **chord**.
- Notes played by the left hand on a piano were classified as **chords.**
- Identified **which notes** were played by the right hand for a **given chord. (P(note|chord))**

🔊 Sample Output-1 🔊 Sample Output-2

Naive Bayes



Keys in Piano

1 2 3 . . . . . 86 87 88

Chords played by left hand

Probability Distribution Table

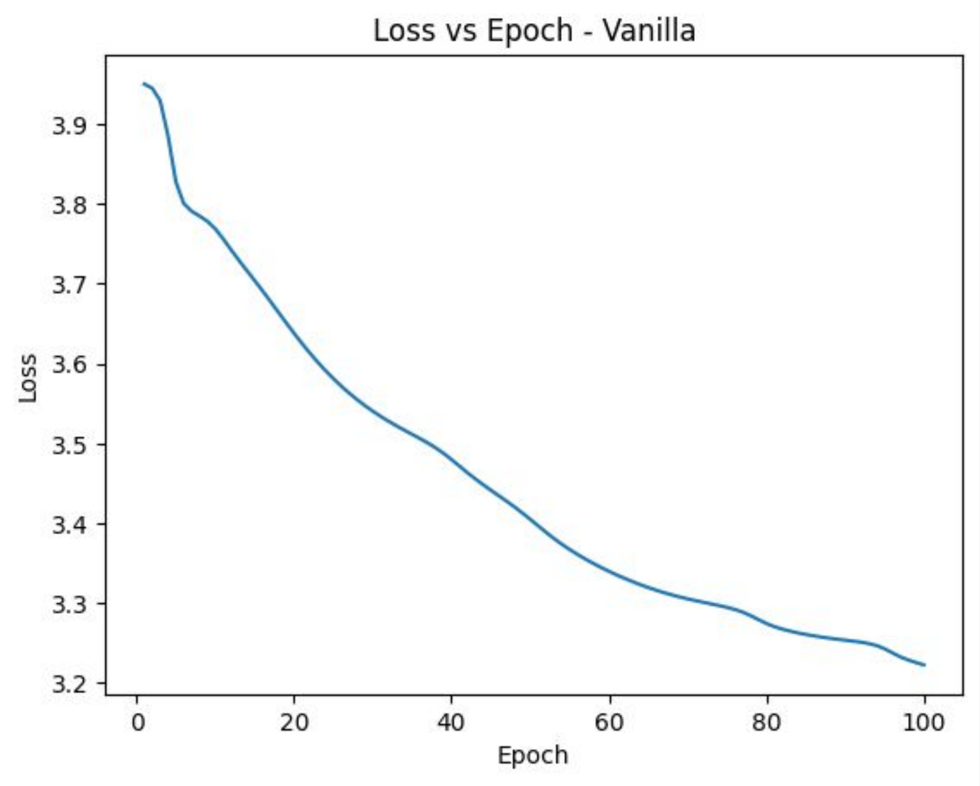
Vanilla Neural Network

- Implemented a basic neural network (NN) for pattern recognition.
- Input for NN, a vector representation of the previous notes played(let’s say 50) is feed into the network.
- The network predicts the next node.
- Showed overfitting.

🔊 Sample Output-1

🔊 Sample Output-2

Vanilla Neural Network



LSTM

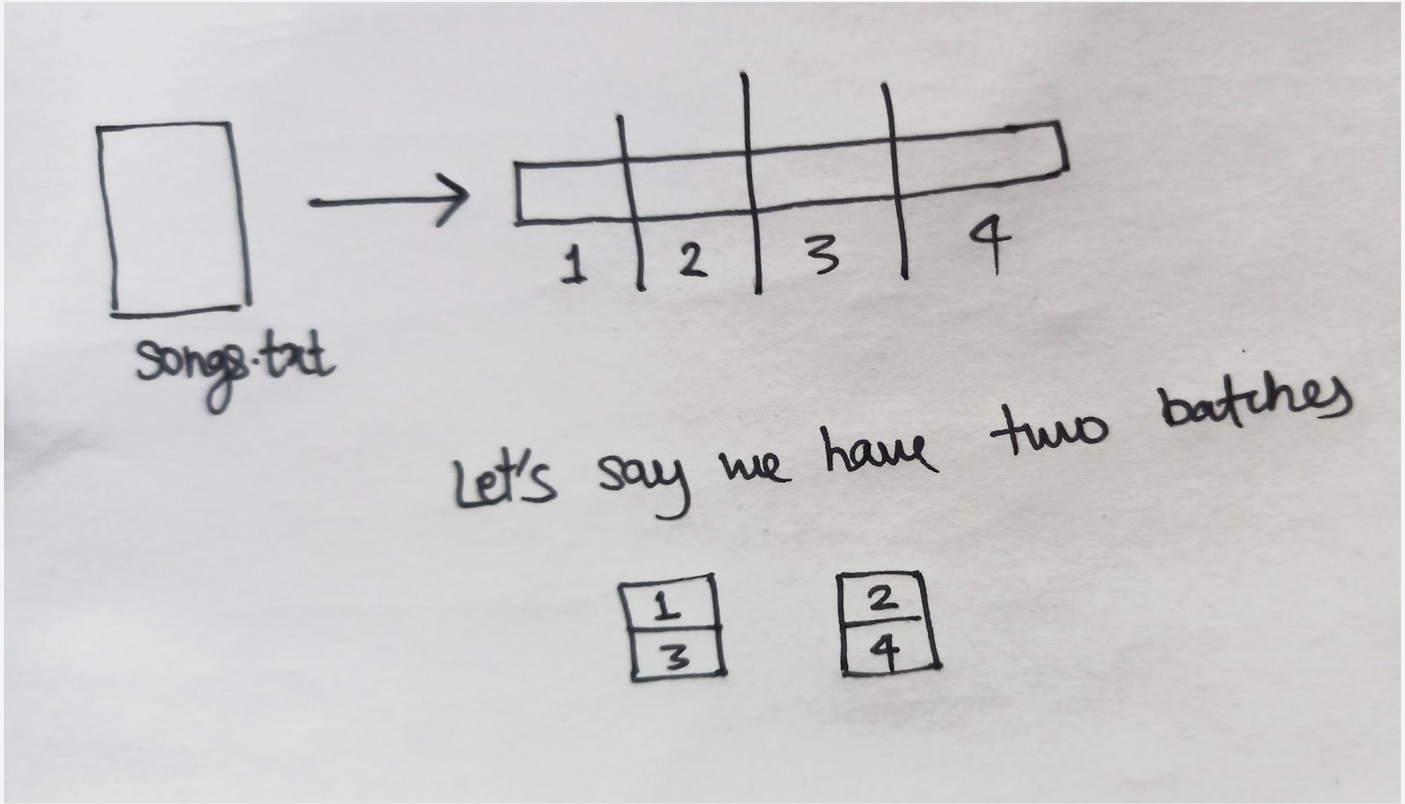
- Generic architecture of LSTM.
- Three gates: input, output, and forget gates.
- Performed much better than Vanilla NN

Sample Output-1

Sample Output-2

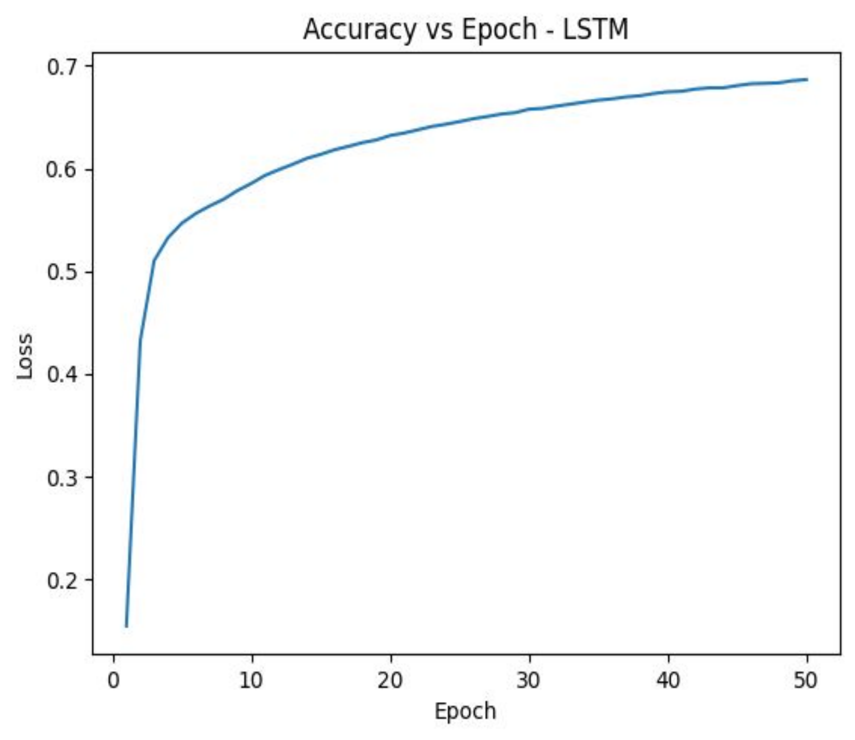
Sample Output-3

LSTM

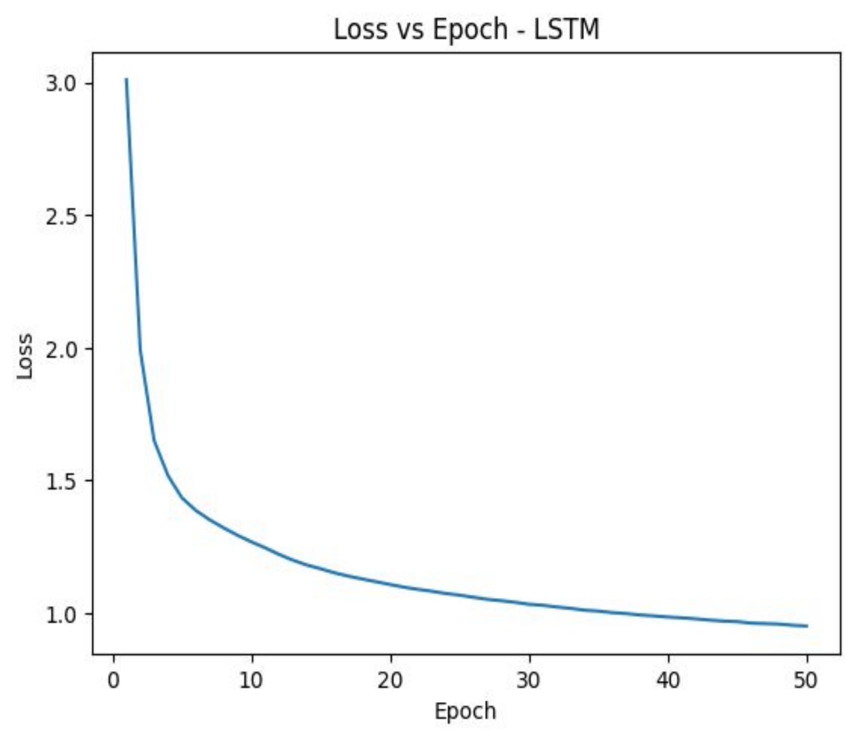


Let's say we have two batches

LSTM



Accuracy vs Epoch - LSTM



Loss vs Epoch - LSTM

Encoder Decoder RNN

- Implemented using GRU layers.
- Output of the last dropout layer in the decoder is then passed through a **linear** and a **sigmoid layer.**

Our Implementation

Markov Chains

- Markov chains are mathematical models used to describe a system that undergoes a sequence of states, where the probability of transitioning from one state to another depends only on the current state and not on the history of states that led to the current state

🔊 Sample Output-1

🔊 Sample Output-2

- Steps Taken:-
    - Preprocessing ABC music files to remove all the music metadata.
    - Extracting individual notes from the music files..like a,b,cl etc.
    - We represent every note as a state of Markov chain, and calculate state **transition probabilities(Matrix)** for going from one state to any other state.
    - Now by giving a random input as start state we can generate a sequence of any length by probabilistic monitoring.

Results from Paper

Results - Naive Bayes

- All the notes played are considered independent.
- The notes are picked randomly from a probability distribution, resulting in a song that may not be soothing to hear.

Results - Vanilla Neural Network

- We noticed that certain notes were unnaturally repeated multiple times.
- It struggles to find pattern sometime and give most played note in the sequence.
- After multiple iterations of training, sometimes notes generated was found in the input dataset.
- It overfits if trained for too many iterations.

Results - LSTM

- Music is soothing to hear. It is hard to detect whether it is human generated or not.
- It finds note/chord pattern in the input sequence.

Results - Markov Chains

- Output generated by markov chains generally contains most frequently occurring notes in the dataset as it is based on probabilistic approach.
- Output produced by changes every time we run for the same state.

Challenges

- Lack of Domain Knowledge.
- Understanding the input format.

Results/ Comparison

- All model wise output can be found here

References

- https://www.tensorflow.org/tutorials/audio/music_generation
- https://www.abcjs.net/abcjs-editor.html
- https://www.online-convert.com/result#j=930801ae-081a-41cd-bab2-0ff56d8f8c00
- https://setosa.io/ev/markov-chains/
- http://mogren.one/publications/2016/c-rnn-gan/mogren2016crnngan.pdf

Thankyou