

# EE626 PROJECT

Music Generation using  
Deep Learning

Devashish Taneja

170102026

Guining Pertin

170102027

Ankit Kumar Saini

170107009

# Idea

Creative arts, the area where new ideas need to be generated is the one field that has been less impacted by AI. This includes graphical art to musical ones, but they have still been affected to a certain extent by the advent of Deep Learning techniques.

We wanted to explore this area of creative music generation using these new techniques in our course project for EE626: Pattern Recognition and Machine Learning.

This idea emerged from us wanting to generate instrument sounds using techniques like GANs but we finally stuck to using ABC notes and RNN models to generate our music due to the time constraint.



# Introduction

Our approach uses Recurrent Neural Network architecture to learn upon the existing music provided in the dataset to finally generating it.

We break down the music into ABC notation, removing the problem of using actual sound wav files. This new data is processed and provided to our neural network to learn from. The recurrent architecture follows the many-to-many scheme where at each step after initialization, the output from previous instance is used as the input for next instance at every step.

Given neural network units with memory, this complex pattern of moving from a certain character to the next can be learned by it and then generate a new music file based on a certain initial input



# Why Neural Networks?

Neural networks used in Deep learning has taken over the AI space in the past few years. From being proved as a universal approximator to being used for generating new images, it has seen rapid development and hence came up as the right choice for our project.

Several online resources was another factor that motivated us to select this technique although it wasn't covered thoroughly in the lectures. The model in our project has been implemented with Tensorflow 2.2 framework.



# Why Recurrent Architecture?

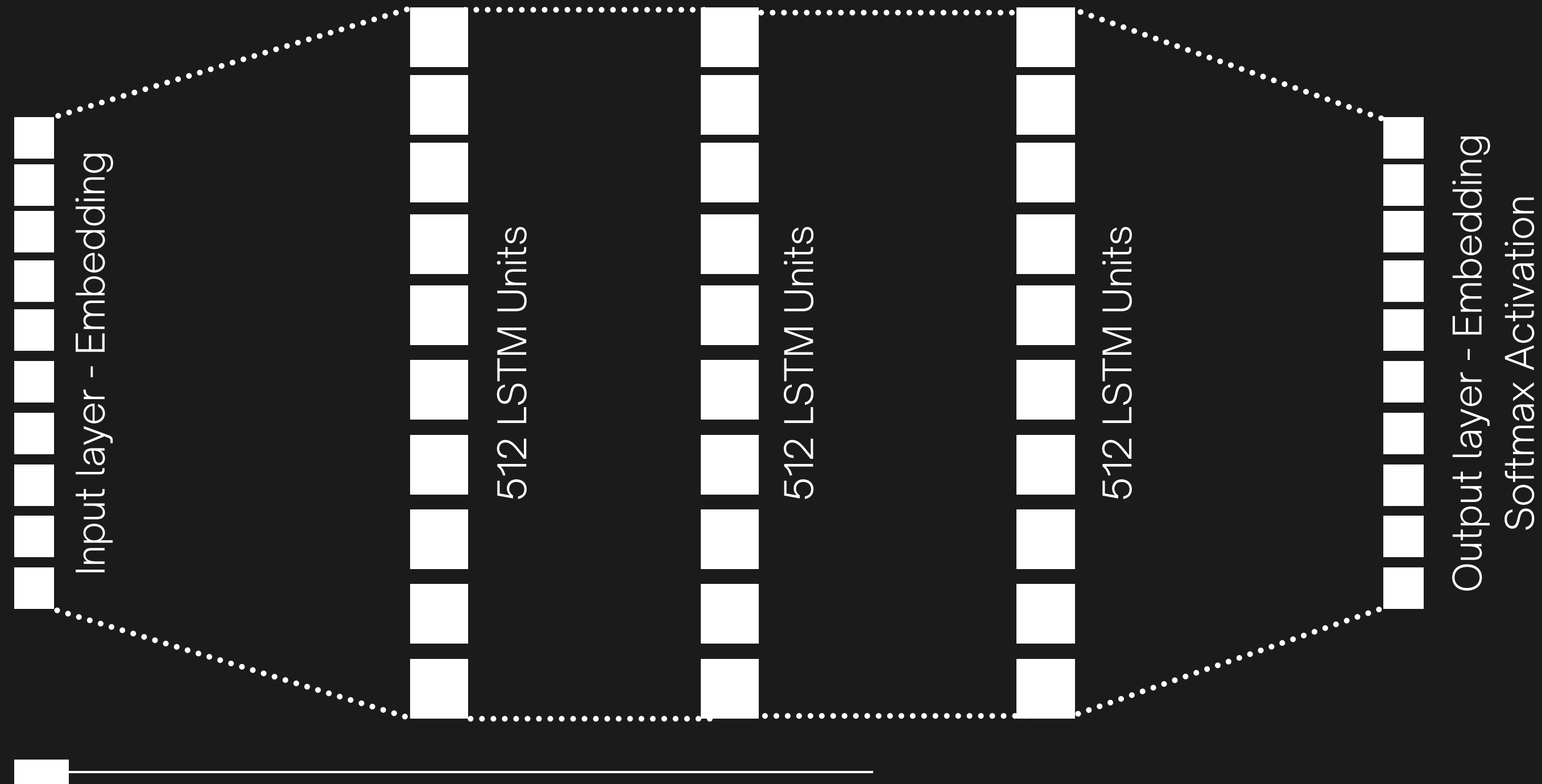
Music is a form of sequential data where this sequence of unique characters play the major role in defining itself. Units with memory element can capture the relationship between the characters at different instances and hence the need for an architecture that supports sequential data input.

LSTM units have been used for our project while the recurrent architecture has been utilized to learn the patterns between different time instances of the data. The final layer uses a Time Distributed layer that helps with this particular architecture. Dropout has also been used for every hidden layer to reduce chances of overfitting when using wide and deep networks.



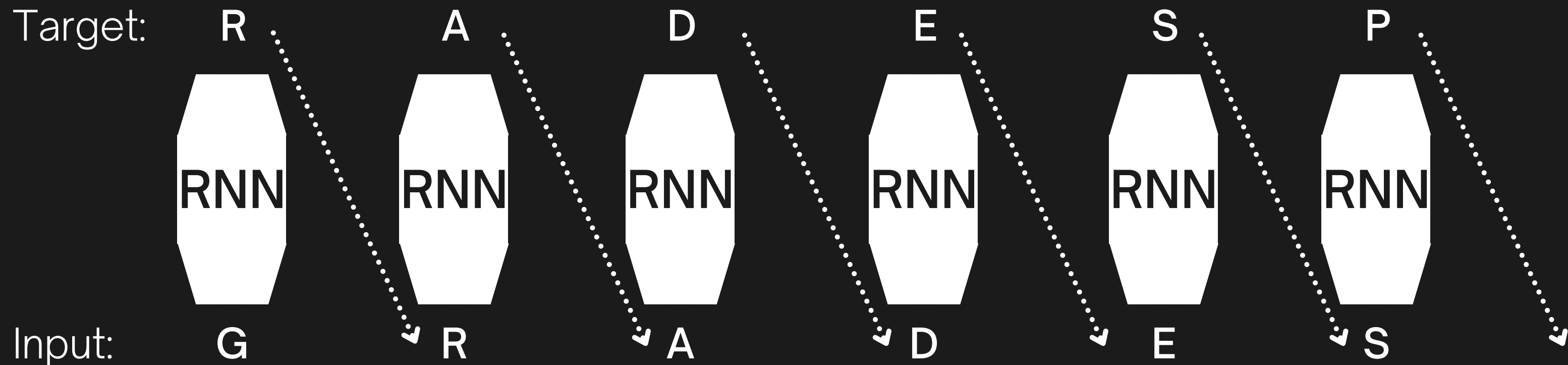
# Neural Network Architecture

We follow the char-RNN architecture by Karpathy with LSTM units:



# Neural Network Architecture

The RNN architecture takes in the current output as the next input in many-to-many sense:



# Data Pre-Processing

Our music data has been taken or converted into ABC format. This data is first cleaned to get the important sections only. The section that carries a form of metadata in ABC format is removed for our case. This final data has different unique characters, most corresponding to different musical notes. This is encoded into one hot vector format initially and provided to the input embedding layer in a certain length defined by the user.

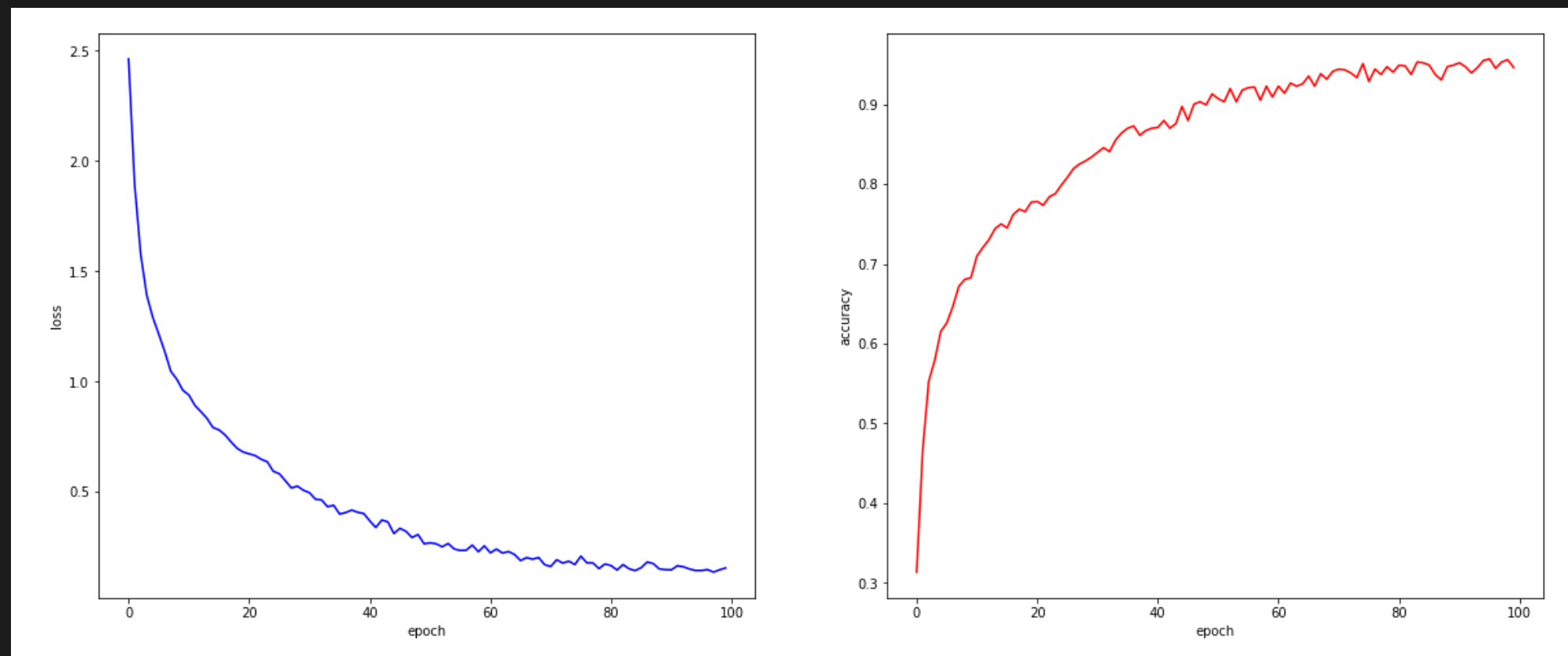
Each instance in the dataset is of certain sequence length and the complete dataset is divided into batches for mini-batch learning.





# Results

The loss and accuracy curves are displayed below. The final ABC notation music generated has also been shown and it has been finally played using midi conversion and software.



```
|:e2e  ecA|AGE  A2B|ced  c2B|ABF  
GFE|D/2B/2A/2G/2  
a/2g/2f/2e/2d/2^c/2|\dB/2c/2d/2B/2  
gB/2c/2d/2B/2|\dB/2c/2d/2B/2  
e/2d/2c/2B/2A/2G/2|B/2G/2B/2d/2g/  
2d/2 b/2a/2g/2f/2e/2d/2^c
```

Although this doesn't make any sense for us, conversion to MIDI shows a completely different picture/sound.

# Conclusion

Our neural network has learned the different ways in which the unique notes connect to other notes based on the previous data it has been provided with. But this is far from generating a completely new music. In our project we explored different ways in which this could be done but we went with the approach that seemed feasible to be trained on our systems.



# References

1. Karpathy's blog - <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>
2. Medium blog - <https://medium.datadriveninvestor.com/music-generation-using-deep-learning-85010fb982e2>
3. Nottingham Music Database - <http://abc.sourceforge.net/NMD/>