# Generation of Music using LSTM over abc notation

-Aman Johar (3035253888), Vashishtha Anushka (3035299404)

COMP3314 Project Report

***Abstract:*** In the past few years, AI has helped significantly in revolutionizing music generation. The long term impact of this technological innovation has made it progressively easier for artists to realize their creative visions, resulting in AI being seen as a powerful tool and partner for the artists. Despite previous study in music generation through machine learning, there is still room to delve into and build sophisticated models. In this work, we use LSTM(RNN) over an "abc" notation to achieve efficient music production.

## I. MOTIVATION

Music is a universal language that can bring people together from all over the world. As emerging technologies help us to communicate better, artificial intelligence is beginning to overtake the music industry. AI is opening up a world where musicians can automate, personalize, and learn from. In this project we will be building on this fact by using neural networks to generate music. First of all, we will use a file having an abc notation as our dataset. This is used most importantly to reduce computation time as compared to more archaic methods involving use of sound recordings as well as it helps in procedurally creating more meaningful sounds. Next, we will feed it to our char-RNN since the dataset is in a text format. Finally, we will train our model which is described further in detail, several time to produce coherent sounds..

## II. BACKGROUND

This sections gives a description of the terminologies and tools which we have used throughout the project.

### Char-RNN

A recurrent neural network is a class of artificial neural networks that make use of sequential information. They are called recurrent because they perform the same function for every single element of a sequence, with the result being dependent on previous computations. Whereas outputs are independent of previous computations in traditional networks. RNN allow feed backward through time. Here, we will use a Long Short Term Memory cells for the RNN. They can efficiently learn via gradient descent. Using a gating mechanism, LSTMs are able to recognise and encode long-term patterns. LSTMs are extremely useful to solve problems where the network has to remember information for a long period of time as is the case in music generation where a character level language model is required.

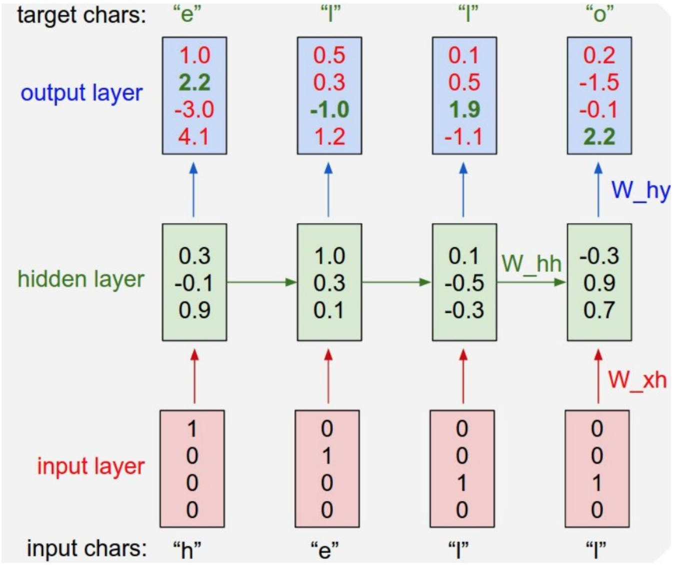



Fig- An example RNN with 4-dimensional input and output layers, and a hidden layer of 3 units (neurons)

## Tensorflow1.0

It is an open source software library for dataflow programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks as in our case. It provides official python API and other languages' APIs as well. Among the applications for which TensorFlow is the foundation, are automated image captioning software, such as DeepDream and RankBrain which handles a substantial number of search queries. In this work, we will use the TensorFlow to create and train the LSTM model. Once the model is trained we will use it to generate the musical notation for our music.

## File with "abc" notation

The file used as dataset for this project contains abc notation which is a shorthand form of musical notation. In basic form it uses the letters A through G to represent the given notes, with other elements used to place added value on these - sharp, flat, the length of the note, key, ornamentation. Nowadays, form of notation is used as an ASCII code that could facilitate the sharing of music. Lines in the first part of the tune notation, beginning with a letter followed by a colon, indicate various aspects of the tune such as the index, when there is more than one tune in a file (X:), the title (T:), the time signature (M:), the default note length (L:), the type of tune (R:) and the key (K:). Lines following the key designation represent the tune.

## Tensorboard

TensorBoard is a visualization suite for the TensorFlow library. It can be used to visualize the TensorFlow graph, plot quantitative metrics about the execution of the graph, and show additional data (e.g. images) that pass through it. Serializing the data can be carried out through the following process. TensorBoard operates by reading TensorFlow events files, which contain summary data that you can generate when running TensorFlow.

# III. RELATED WORK

Algorithmic music composition has developed a lot in the last few years, but the idea has a long history. Musikalisches Würfelspiel, a game that generates short piano compositions from fragments, with choices made by dice is a very rudimentary approach.

Markov chains can also be used to generate new musical compositions and take the motivations behind the dice game a step further, in two ways. First, Markov chains can be built from existing material rather than needing fragments explicitly composed as interchangeable components. Second, instead of assuming fragments have equal probabilities, Markov chains encode the variation in probabilities with respect to context. Iannis Xenakis used Markov chains in his 1958 compositions, "Analogique" where he used transition matrices that define the probabilities of certain notes being produced.

In 1981 David Cope combined Markov chains and other techniques into a semi-automatic system called Emmy which is most famous for learning from and imitating other composers. While Markov chains trained on a set of compositions can only produce subsequences that also exist in the original data RNNs attempt to extrapolate beyond those exact subsequences.

In 1989, Peter M. Todd, Michael C. Mozer and others used RNNs but were limited by their short-term coherence.

In 2002 Doug Eck switched to LSTM which was able to play blues with good timing and proper structure. Doug as part of Google's magenta has applied LSTM-based approaches to drum pattern generation, melody generation, and polyphonic music generation. They've built systems that improvise duets with human performers, and tools that generate expressive dynamics and timing along with the polyphonic compositions.

All these methods give different insight to the process of music generation but I believe that the method used in the project takes it one more notch higher by introducing the dataset in an abc format. As described before, this dataset helps in reducing computation time and increasing the computation and processing efficacy as no conversion of sound recordings to vector is required. This step is very cumbersome and takes away meaning from the main process of music generation, hence we totally avoid it. Furthermore, abc notation helps in procedurally creating more meaningful sounds.

## IV. METHODS

### Dataset

The Nottingham Music Database used in this project and maintained by Eric Foxley contains over 1000 Folk Tunes stored in a special text format that is abc. Using NMD2ABC, a program written by Jay Glanville and various other perl scripts, the bulk of this database has been converted to abc notation. More recently, Seymour Shlien has edited the files to correct a few problems. We are using all of the folk tunes to train our model for this project

### Word2Vec

A common thing we do in natural language processing and machine learning is to map words to numeric values in order for the machine to understand it. For this project we used word embedding to do the above. The embedding method we used was Word2Vec with a CBOW model. In CBOW, all the examples with the target word as target are fed into the networks, and taking the average of the extracted hidden layer

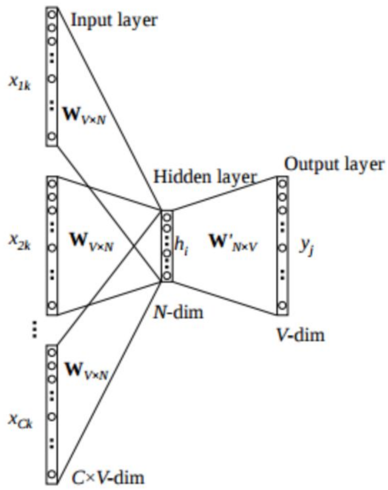



Fig- CBOW

### Char-RNN

Once we have encoded our dataset with numerical values, we move on and feed the dataset to the RNN. The basic idea here is to predict the distributions of what the next character could be given a character fed into the RNN. We perform parameter update which nudges the weight into the right direction. In our project we track our progress using train_loss which should reduce with every epoch. We repeat this process over and over until the RNN converges and the predictions starting becoming consistent with the training data.

Our implementation starts with implementing a Char-RNN which is used to generate new characters based on existing sequences(the dataset- in this case the abc notations). We have based our Char-RNN implementation on Andrej Karpathy's Char-RNN network. Using MultiRNN cells (multiple layers of LSTMs give better power to the model) and epoch=50 (hyperparameter tuning) we have been able to achieve far more intelligible music notations then the previous work could have ever had.

We have used used the standard Softmax classifier on the output layer. We are training our RNN with mini batch Stochastic descent and using Adam optimizer to stabilize the weight updates

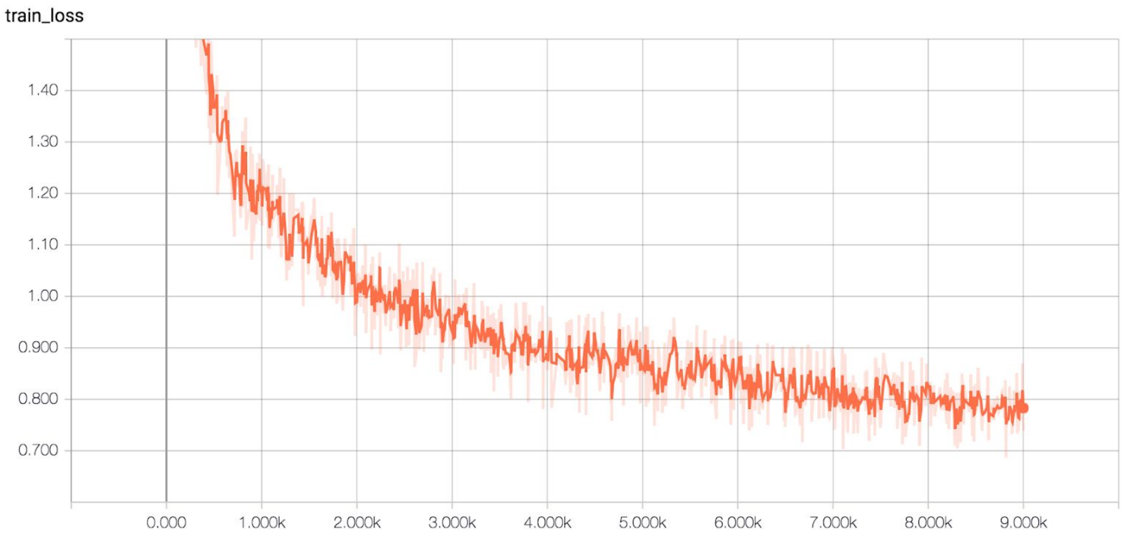
## V. RESULTS AND DISCUSSION

Our aim is to minimize the train_loss as much as possible. The train_loss in our project is defined by the following function:

$$\text{loss} = -\frac{1}{N}\sum_{i=1}^{N} \ln p_{\text{target}_i}$$

This is the average negative log probability of the target characters

We have monitored the value of train_loss during the training process using tensorboard and tensor logs and have produced the following result:




Graph-> y-axis:train_loss    x-axis: (epoch * total_number_of_batches) + currently trained batch

We can see that as our training progresses, the value of train_loss decreases. This shows that our model is getting better at making predictions with every stage of training.

We trained our model for epoch=50 and approximately had 180 batches to train keeping size of each batch to be 50 and sequence length per batch to be 50 as well. These hyperparameters gave us our best result achievable (given the resources we had)

<u>Resulting ABC notations</u>

Upon training our model according to the above constraints, it was able to produce some intelligible ABC notation files. Here are a few examples of the ABC notations produced by our model:

```
X: 13
T:Radmy Ball
% Nottingham Music Database
P:AAB
S:Chris Dewhurst 19
M:4/4
L:1/8
R:Hornpipe
K:D
P:A
cf|"D"f2ed f2f2|"Em"g2f2 B2^d2|"Em"efgg "D"fede|"G"dcBA BdcB|"gbe/2^f/2|"Em"edc|"Em"e2f|
"Am"e3/2d/2c/2B/2|"Am"A3/2B/2A|"D7"A3/2F/2A|"G"B3/2c/2d|"Em"g2e|"F"f2a|"Bm"f3|"C/e"ed2 "Am"f2e|"D7"d2c|
"G"dBG|"Am"F2A "E7"FED|"D7"A2G "Am"G2G|"G"F2G "D7/a"AFG|
"C"c2e "F" a3|"Bb"dcB "Dm"A2F|"G"G2D G2B|"Am"e3 "G7"cBA|"F#m"B4 ||
```

```
X: 1
T:Dirne's Reel
% Nottingham Music Database
S:Trad
M:3/4
L:1/4
K:Am
P:A
g|"Am"f/2e/2d/2c/2d/2c/2|"G"BB/2G/2F/2B/2|"Em"B/2A/2G/2E/2F/2G/2|"Am"A/2B/2A/2F/2G/2E/2:|
"D"D/2E/2D/2D/2D/2E/2 d/2c/2A/2B/2A/2F/2||/2D/2D/2F/2A/2D/2 GD/2D/2|\
"G"E/2DX: (F7
|:"b"fg f3/2a/2|"C"ge^d ece|"G"fg2 gab|"D7"dcd d^c"f#"d\
||
```

```
X: 12
T:The Welm Brband P1y
% Nottingham Music Database
S:Chris Dewhurst 1955, via Phil Rowe
M:6/8
K:Gm
D|"G"BGB GGB|"C"c2A G3|"C"(3cBcA G2A|"Cm"dcB ecG|"D7"F2D cFD|
"G"DGB dBG|"G"GFG "E7"BAG|"Am"Ace "Dm"fed|"Em"edB "G"G\
:|
K:Am
P:/8C/8C/8Ep8g/8f/8"A7"d/2(3e d/2+ec cAA|"Em"e3 g3:|
P:B
(3G/2G/2B/2|"A"A2A "C"[c3g3|"D"F2A B3-|2c2 B2A|A2B A2B|"A/c"c2A "E7"^G2A|
"Am"cBA c3|"G"dBG "3"G3:|
```

```
X: 67
T:The The Fluenssamle Sireby
% Nottingham Music Database
S:Mick Peat
M:4/4
L:1/4
K:D
P:A
:|:A|"D"df dg|"A""f#"fd dB|"Em"d2 fe|"D"d2 -d2|\
"A7"c/2B/2=c fe|"F#7"Ag2e|"D"f2 f3/2f/2|
"Em"ee "E7"ge|"D"a2 fA|"D"d2 d2|"C"eg gc|A3g|"G"d4|"Em"g2 -e/2d/2c|
"G"BG "D"FE|"D"F2 FG|"D"F3/2G/2 Ad|"Bm"de dc|"G"B/2dB/2 Bc|"C"dc cB-|"C"eG2G|B2c|B4|"Am"c3/2d/2 ec|\
"G"G4-|
"G"B3/2c/2 -d2|"C"=ee c3/2a/2|"G"B/2^A/2B/2g/2 gf/2d/2|\
"D7"d2 c2| "G"G2-|D/2D/2E/2d/2 =B3/2d/2|
"G"dd d||
K:G
"G7"b2 "A"a2|"G"g4|"G"a/2g/2f/2e/2 de|"D7/b"d2 "G"AG|"C"E/2D/2E "G"Dd|
```

## VI. CONCLUSION AND FUTURE WORK

We can see that training an RNN/LSTM on ABC notation files can successfully reproduce different combinations of intelligible ABC notations. Due to the limits of available resources, we were able to produce only a few fully usable ABC notations. The model we used is promising and with enough training it can repeatedly produce proper ABC notations which can be converted into melodies with the help of a conversion tool.

LSTMs manage to maintain long-term consistency better than a standard RNN or Markov chain, there is still a gap between generating short phrases and generating an entire composition; something that has not yet been bridged without lots of tricks and hand-tuning. Startups like Aiva and others are trying to fill this space of on-demand, hand-tuned formulaic generative music. Some going so far so as to produce entire pop albums as marketing. Big companies are getting in on the action, too. François Pachet, formerly at Sony Computer Science Laboratories and now at Spotify, has been working with algorithmic music for some time, from his Continuator to the more recent Flow machines.

## CONTRIBUTION

All the team members have contributed equally in the process of music generation in all parts as well as the fundamental idea involved.

## REFERENCES

Walshaw, Chris. "abc music notation :: introduction". abcnotation.com. Retrieved March 1, 2008.

https://www.tensorflow.org

McDoanald K. “Neural Nets for generating muysic, August 26 2017 Retrieved from: ”https://medium.com/artists-and-machine-intelligence/neural-nets-for-generating-music-f46dffac21c0

Allwright J. “ABC version of Nottingham Music Database” Retrieved from: http://abc.sourceforge.net/NMD/

Karpathy A. “The unreasonable Effectiveness of Recurrent Neural Networks”, May 21 2015, Retrieve from: http://karpathy.github.io/2015/05/21/rnn-effectiveness/

https://www.tensorflow.org/programmers_guide/summaries_and_tensorboard

Araoz M. “Training a Recurrent neural Network to Compose music”, Feb. 2 2016, Retrieved from https://maraoz.com/2016/02/02/abc-rnn/

Skuli S. ”How to generate music using a LSTM neural network in keras”, December 7 2017, Retrieved from: https://towardsdatascience.com/how-to-generate-music-using-a-lstm-neural-network-in-keras-68786834d4c5