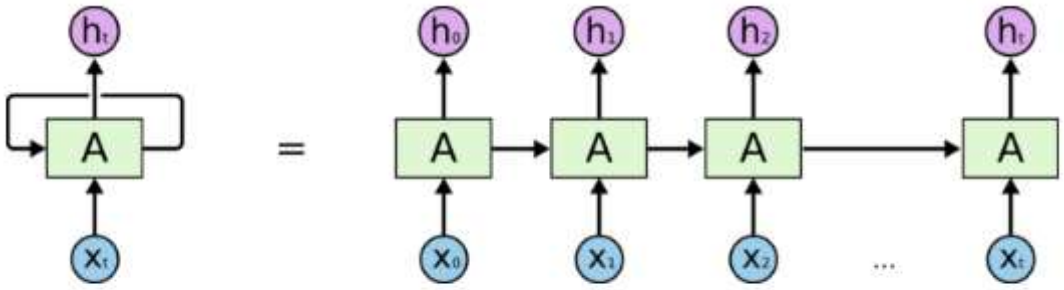
Music is an Art and a Universal language.

Approach Using Long Short Term Memory (LSTM) Model : Long Short Term Memory Model, popularly known as LSTM, is a variant of Recurrent Neural Networks (RNNs) that is capable of capturing the long term dependencies in the input sequence. LSTM has a wide range of applications in Sequence-to-Sequence modeling tasks like Speech Recognition, Text Summarization, Video Classification, and so on.



An unrolled recurrent neural network.

The Workflow of model:

- Input is fed into a causal 1D convolution
- The output is then fed to 2 different dilated 1D convolution layers with sigmoid and tanh activations
- The element-wise multiplication of 2 different activation values results in a skip connection
- And the element-wise addition of a skip connection and output of causal 1D results in the residual

In [ ]:
```
%pip install pandas matplotlib scipy seaborn tensorflow music21 keras numpy scikit-
```

In [1]:
```
from music21 import *
from sklearn.model_selection import train_test_split
```

In [21]:
```
#defining function to read MIDI files
def read_midi(file):

    print("Loading Music File:",file)

    notes=[]
    notes_to_parse = None

    #parsing a midi file
    midi = converter.parse(file)

    #grouping based on different instruments
    s2 = instrument.partitionByInstrument(midi)

    #Looping over all the instruments
    for part in s2.parts:
```

```
        #select elements of only piano
        if 'Piano' in str(part):

            notes_to_parse = part.recurse()

            #finding whether a particular element is note or a chord
            for element in notes_to_parse:

                #note
                if isinstance(element, note.Note):
                    notes.append(str(element.pitch))

                #chord
                elif isinstance(element, chord.Chord):
                    notes.append('.'.join(str(n) for n in element.normalOrder))

    return np.array(notes)
```

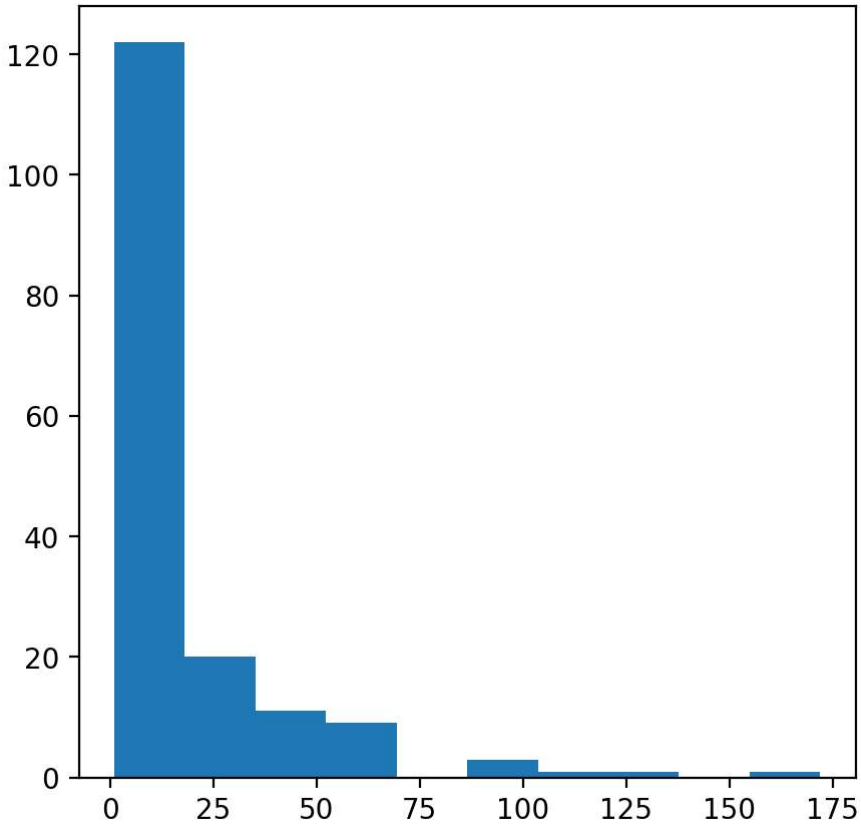
In [22]:
```
#for listing down the file names
import os

#Array Processing
import numpy as np

#specify the path
path='data/'

#read all the filenames
files=[i for i in os.listdir(path) if i.endswith(".mid")]

#reading each midi file
notes_array = np.array([read_midi(path+i) for i in files], dtype=object)

#converting 2D array into 1D array
notes_ = [element for note_ in notes_array for element in note_]

#No. of unique notes
unique_notes = list(set(notes_))
print(len(unique_notes))
```

```
Loading Music File: data/1.mid
Loading Music File: data/2.mid
Loading Music File: data/3.mid
Loading Music File: data/4.mid
Loading Music File: data/5.mid
168
```

In [23]:
```
#importing library
from collections import Counter

#computing frequency of each note
freq = dict(Counter(notes_))

#library for visualiation
import matplotlib.pyplot as plt

#consider only the frequencies
no=[count for _,count in freq.items()]

#set the figure size
plt.figure(figsize=(5,5))
```

```
#plot
plt.hist(no)
```

Out[23]:
```
(array([122.,  20.,  11.,   9.,   0.,   3.,   1.,   1.,   0.,   1.]),
 array([  1. ,  18.1,  35.2,  52.3,  69.4,  86.5, 103.6, 120.7, 137.8,
        154.9, 172. ]),
 <BarContainer object of 10 artists>)
```



In [24]:
```
frequent_notes = [note_ for note_, count in freq.items() if count>=20]
print(len(frequent_notes))
```

```
45
```

In [25]:
```
new_music=[]

for notes in notes_array:
    temp=[]
    for note_ in notes:
        if note_ in frequent_notes:
            temp.append(note_)
    new_music.append(temp)

new_music = np.array(new_music,dtype=object)
```

In [26]:
```
no_of_timesteps = 32
x = []
y = []

for note_ in new_music:
    for i in range(0, len(note_) - no_of_timesteps, 1):

        #preparing input and output sequences
        input_ = note_[i:i + no_of_timesteps]
        output = note_[i + no_of_timesteps]
```

```
        x.append(input_)
        y.append(output)

x=np.array(x)
y=np.array(y)
```

In [27]:
```
unique_x = list(set(x.ravel()))
x_note_to_int = dict((note_, number) for number, note_ in enumerate(unique_x))
```

In [28]:
```
#preparing input sequences
x_seq=[]
for i in x:
    temp=[]
    for j in i:
        #assigning unique integer to every note
        temp.append(x_note_to_int[j])
    x_seq.append(temp)

x_seq = np.array(x_seq)
```

In [29]:
```
unique_y = list(set(y))
y_note_to_int = dict((note_, number) for number, note_ in enumerate(unique_y))
y_seq=np.array([y_note_to_int[i] for i in y])
```

In [30]:
```
x_tr, x_val, y_tr, y_val = train_test_split(x_seq,y_seq,test_size=0.2,random_state=
```

In [31]:
```
def lstm():
  model = Sequential()
  model.add(LSTM(128,return_sequences=True))
  model.add(LSTM(128))
  model.add(Dense(256))
  model.add(Activation('relu'))
  model.add(Dense(n_vocab))
  model.add(Activation('softmax'))
  model.compile(loss='sparse_categorical_crossentropy', optimizer='adam')
  return model
```

In [32]:
```
from keras.layers import *
from keras.models import *
from keras.callbacks import *
import keras.backend as K

K.clear_session()
model = Sequential()

#embedding layer
model.add(Embedding(len(unique_x), 100, input_length=32,trainable=True))

model.add(Conv1D(64,3, padding='causal',activation='relu'))
model.add(Dropout(0.2))
model.add(MaxPool1D(2))

model.add(Conv1D(128,3,activation='relu',dilation_rate=2,padding='causal'))
model.add(Dropout(0.2))
model.add(MaxPool1D(2))

model.add(Conv1D(256,3,activation='relu',dilation_rate=4,padding='causal'))
model.add(Dropout(0.2))
model.add(MaxPool1D(2))
```

```
#model.add(Conv1D(256,5,activation='relu'))
model.add(GlobalMaxPool1D())

model.add(Dense(256, activation='relu'))
model.add(Dense(len(unique_y), activation='softmax'))

model.compile(loss='sparse_categorical_crossentropy', optimizer='adam')

model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding (Embedding)       (None, 32, 100)           4500

 conv1d (Conv1D)             (None, 32, 64)            19264

 dropout (Dropout)           (None, 32, 64)            0

 max_pooling1d (MaxPooling1D  (None, 16, 64)           0
 )

 conv1d_1 (Conv1D)           (None, 16, 128)           24704

 dropout_1 (Dropout)         (None, 16, 128)           0

 max_pooling1d_1 (MaxPooling  (None, 8, 128)           0
 1D)

 conv1d_2 (Conv1D)           (None, 8, 256)            98560

 dropout_2 (Dropout)         (None, 8, 256)            0

 max_pooling1d_2 (MaxPooling  (None, 4, 256)           0
 1D)

 global_max_pooling1d (Globa  (None, 256)              0
 lMaxPooling1D)

 dense (Dense)               (None, 256)               65792

 dense_1 (Dense)             (None, 45)                11565

=================================================================
Total params: 224,385
Trainable params: 224,385
Non-trainable params: 0
_________________________________________________________________
```

In [33]:
```
mc=ModelCheckpoint('best_model.h5', monitor='val_loss', mode='min', save_best_only=
```

In [34]:
```
history = model.fit(np.array(x_tr),np.array(y_tr),batch_size=128,epochs=100, valida
```

```
Epoch 1/100
12/13 [==========================>...] - ETA: 0s - loss: 3.7538
Epoch 1: val_loss improved from inf to 3.65195, saving model to best_model.h5
13/13 [==============================] - 3s 118ms/step - loss: 3.7482 - val_loss:
3.6520
Epoch 2/100
12/13 [==========================>...] - ETA: 0s - loss: 3.6099
Epoch 2: val_loss improved from 3.65195 to 3.58974, saving model to best_model.h5
13/13 [==============================] - 1s 85ms/step - loss: 3.6080 - val_loss:
3.5897
Epoch 3/100
12/13 [==========================>...] - ETA: 0s - loss: 3.5368
Epoch 3: val_loss improved from 3.58974 to 3.53650, saving model to best_model.h5
13/13 [==============================] - 1s 85ms/step - loss: 3.5408 - val_loss:
3.5365
Epoch 4/100
12/13 [==========================>...] - ETA: 0s - loss: 3.4556
Epoch 4: val_loss improved from 3.53650 to 3.43553, saving model to best_model.h5
13/13 [==============================] - 1s 87ms/step - loss: 3.4555 - val_loss:
3.4355
Epoch 5/100
12/13 [==========================>...] - ETA: 0s - loss: 3.3108
Epoch 5: val_loss improved from 3.43553 to 3.31640, saving model to best_model.h5
13/13 [==============================] - 1s 85ms/step - loss: 3.3125 - val_loss:
3.3164
Epoch 6/100
12/13 [==========================>...] - ETA: 0s - loss: 3.1501
Epoch 6: val_loss improved from 3.31640 to 3.15188, saving model to best_model.h5
13/13 [==============================] - 1s 85ms/step - loss: 3.1414 - val_loss:
3.1519
Epoch 7/100
12/13 [==========================>...] - ETA: 0s - loss: 2.9793
Epoch 7: val_loss improved from 3.15188 to 3.02652, saving model to best_model.h5
13/13 [==============================] - 1s 84ms/step - loss: 2.9836 - val_loss:
3.0265
Epoch 8/100
13/13 [==============================] - ETA: 0s - loss: 2.7950
Epoch 8: val_loss improved from 3.02652 to 2.87216, saving model to best_model.h5
13/13 [==============================] - 1s 100ms/step - loss: 2.7950 - val_loss:
2.8722
Epoch 9/100
13/13 [==============================] - ETA: 0s - loss: 2.6201
Epoch 9: val_loss improved from 2.87216 to 2.78135, saving model to best_model.h5
13/13 [==============================] - 1s 90ms/step - loss: 2.6201 - val_loss:
2.7814
Epoch 10/100
12/13 [==========================>...] - ETA: 0s - loss: 2.5176
Epoch 10: val_loss improved from 2.78135 to 2.71436, saving model to best_model.h5
13/13 [==============================] - 1s 86ms/step - loss: 2.5055 - val_loss:
2.7144
Epoch 11/100
12/13 [==========================>...] - ETA: 0s - loss: 2.3936
Epoch 11: val_loss improved from 2.71436 to 2.60972, saving model to best_model.h5
13/13 [==============================] - 1s 85ms/step - loss: 2.4011 - val_loss:
2.6097
Epoch 12/100
12/13 [==========================>...] - ETA: 0s - loss: 2.2636
Epoch 12: val_loss improved from 2.60972 to 2.53014, saving model to best_model.h5
13/13 [==============================] - 1s 86ms/step - loss: 2.2598 - val_loss:
2.5301
Epoch 13/100
```

```
12/13 [=========================>...] - ETA: 0s - loss: 2.1691
Epoch 13: val_loss improved from 2.53014 to 2.44950, saving model to best_model.h5
13/13 [==============================] - 1s 86ms/step - loss: 2.1594 - val_loss:
2.4495
Epoch 14/100
12/13 [=========================>...] - ETA: 0s - loss: 2.0346
Epoch 14: val_loss improved from 2.44950 to 2.36786, saving model to best_model.h5
13/13 [==============================] - 1s 92ms/step - loss: 2.0440 - val_loss:
2.3679
Epoch 15/100
13/13 [==============================] - ETA: 0s - loss: 1.9878
Epoch 15: val_loss improved from 2.36786 to 2.33637, saving model to best_model.h5
13/13 [==============================] - 1s 100ms/step - loss: 1.9878 - val_loss:
2.3364
Epoch 16/100
13/13 [==============================] - ETA: 0s - loss: 1.9398
Epoch 16: val_loss improved from 2.33637 to 2.27779, saving model to best_model.h5
13/13 [==============================] - 1s 101ms/step - loss: 1.9398 - val_loss:
2.2778
Epoch 17/100
13/13 [==============================] - ETA: 0s - loss: 1.8482
Epoch 17: val_loss improved from 2.27779 to 2.22847, saving model to best_model.h5
13/13 [==============================] - 1s 100ms/step - loss: 1.8482 - val_loss:
2.2285
Epoch 18/100
12/13 [=========================>...] - ETA: 0s - loss: 1.7845
Epoch 18: val_loss improved from 2.22847 to 2.12062, saving model to best_model.h5
13/13 [==============================] - 1s 86ms/step - loss: 1.7921 - val_loss:
2.1206
Epoch 19/100
12/13 [=========================>...] - ETA: 0s - loss: 1.7045
Epoch 19: val_loss improved from 2.12062 to 2.07617, saving model to best_model.h5
13/13 [==============================] - 1s 86ms/step - loss: 1.7008 - val_loss:
2.0762
Epoch 20/100
12/13 [=========================>...] - ETA: 0s - loss: 1.6067
Epoch 20: val_loss improved from 2.07617 to 2.02056, saving model to best_model.h5
13/13 [==============================] - 1s 90ms/step - loss: 1.6173 - val_loss:
2.0206
Epoch 21/100
12/13 [=========================>...] - ETA: 0s - loss: 1.5773
Epoch 21: val_loss improved from 2.02056 to 1.96283, saving model to best_model.h5
13/13 [==============================] - 1s 85ms/step - loss: 1.5837 - val_loss:
1.9628
Epoch 22/100
12/13 [=========================>...] - ETA: 0s - loss: 1.5044
Epoch 22: val_loss improved from 1.96283 to 1.95969, saving model to best_model.h5
13/13 [==============================] - 1s 86ms/step - loss: 1.5069 - val_loss:
1.9597
Epoch 23/100
12/13 [=========================>...] - ETA: 0s - loss: 1.4355
Epoch 23: val_loss improved from 1.95969 to 1.88115, saving model to best_model.h5
13/13 [==============================] - 1s 86ms/step - loss: 1.4426 - val_loss:
1.8811
Epoch 24/100
12/13 [=========================>...] - ETA: 0s - loss: 1.3788
Epoch 24: val_loss improved from 1.88115 to 1.85905, saving model to best_model.h5
13/13 [==============================] - 1s 86ms/step - loss: 1.3799 - val_loss:
1.8590
Epoch 25/100
12/13 [=========================>...] - ETA: 0s - loss: 1.3246
```

```
Epoch 25: val_loss improved from 1.85905 to 1.81650, saving model to best_model.h5
13/13 [==============================] - 1s 86ms/step - loss: 1.3325 - val_loss:
1.8165
Epoch 26/100
12/13 [=========================>...] - ETA: 0s - loss: 1.2819
Epoch 26: val_loss improved from 1.81650 to 1.73318, saving model to best_model.h5
13/13 [==============================] - 1s 89ms/step - loss: 1.2917 - val_loss:
1.7332
Epoch 27/100
12/13 [=========================>...] - ETA: 0s - loss: 1.2279
Epoch 27: val_loss improved from 1.73318 to 1.72173, saving model to best_model.h5
13/13 [==============================] - 1s 85ms/step - loss: 1.2325 - val_loss:
1.7217
Epoch 28/100
13/13 [==============================] - ETA: 0s - loss: 1.2094
Epoch 28: val_loss improved from 1.72173 to 1.69885, saving model to best_model.h5
13/13 [==============================] - 1s 105ms/step - loss: 1.2094 - val_loss:
1.6989
Epoch 29/100
13/13 [==============================] - ETA: 0s - loss: 1.1754
Epoch 29: val_loss improved from 1.69885 to 1.64359, saving model to best_model.h5
13/13 [==============================] - 2s 124ms/step - loss: 1.1754 - val_loss:
1.6436
Epoch 30/100
13/13 [==============================] - ETA: 0s - loss: 1.1124
Epoch 30: val_loss improved from 1.64359 to 1.58297, saving model to best_model.h5
13/13 [==============================] - 1s 116ms/step - loss: 1.1124 - val_loss:
1.5830
Epoch 31/100
13/13 [==============================] - ETA: 0s - loss: 1.0724
Epoch 31: val_loss did not improve from 1.58297
13/13 [==============================] - 2s 146ms/step - loss: 1.0724 - val_loss:
1.5892
Epoch 32/100
13/13 [==============================] - ETA: 0s - loss: 1.0088
Epoch 32: val_loss improved from 1.58297 to 1.54120, saving model to best_model.h5
13/13 [==============================] - 2s 170ms/step - loss: 1.0088 - val_loss:
1.5412
Epoch 33/100
13/13 [==============================] - ETA: 0s - loss: 0.9611
Epoch 33: val_loss did not improve from 1.54120
13/13 [==============================] - 2s 151ms/step - loss: 0.9611 - val_loss:
1.5624
Epoch 34/100
13/13 [==============================] - ETA: 0s - loss: 0.9656
Epoch 34: val_loss improved from 1.54120 to 1.52897, saving model to best_model.h5
13/13 [==============================] - 2s 185ms/step - loss: 0.9656 - val_loss:
1.5290
Epoch 35/100
13/13 [==============================] - ETA: 0s - loss: 0.8887
Epoch 35: val_loss improved from 1.52897 to 1.47095, saving model to best_model.h5
13/13 [==============================] - 2s 182ms/step - loss: 0.8887 - val_loss:
1.4710
Epoch 36/100
13/13 [==============================] - ETA: 0s - loss: 0.8605
Epoch 36: val_loss did not improve from 1.47095
13/13 [==============================] - 2s 175ms/step - loss: 0.8605 - val_loss:
1.4865
Epoch 37/100
13/13 [==============================] - ETA: 0s - loss: 0.8381
Epoch 37: val_loss improved from 1.47095 to 1.44679, saving model to best_model.h5
```

```
13/13 [==============================] - 2s 153ms/step - loss: 0.8381 - val_loss:
1.4468
Epoch 38/100
13/13 [==============================] - ETA: 0s - loss: 0.8096
Epoch 38: val_loss did not improve from 1.44679
13/13 [==============================] - 2s 120ms/step - loss: 0.8096 - val_loss:
1.4697
Epoch 39/100
13/13 [==============================] - ETA: 0s - loss: 0.8017
Epoch 39: val_loss improved from 1.44679 to 1.43537, saving model to best_model.h5
13/13 [==============================] - 2s 127ms/step - loss: 0.8017 - val_loss:
1.4354
Epoch 40/100
13/13 [==============================] - ETA: 0s - loss: 0.7794
Epoch 40: val_loss did not improve from 1.43537
13/13 [==============================] - 2s 130ms/step - loss: 0.7794 - val_loss:
1.4443
Epoch 41/100
13/13 [==============================] - ETA: 0s - loss: 0.7605
Epoch 41: val_loss did not improve from 1.43537
13/13 [==============================] - 2s 157ms/step - loss: 0.7605 - val_loss:
1.4365
Epoch 42/100
13/13 [==============================] - ETA: 0s - loss: 0.7462
Epoch 42: val_loss improved from 1.43537 to 1.43155, saving model to best_model.h5
13/13 [==============================] - 2s 147ms/step - loss: 0.7462 - val_loss:
1.4316
Epoch 43/100
13/13 [==============================] - ETA: 0s - loss: 0.6969
Epoch 43: val_loss improved from 1.43155 to 1.41248, saving model to best_model.h5
13/13 [==============================] - 1s 101ms/step - loss: 0.6969 - val_loss:
1.4125
Epoch 44/100
13/13 [==============================] - ETA: 0s - loss: 0.6743
Epoch 44: val_loss improved from 1.41248 to 1.40050, saving model to best_model.h5
13/13 [==============================] - 1s 111ms/step - loss: 0.6743 - val_loss:
1.4005
Epoch 45/100
13/13 [==============================] - ETA: 0s - loss: 0.6658
Epoch 45: val_loss improved from 1.40050 to 1.39053, saving model to best_model.h5
13/13 [==============================] - 1s 105ms/step - loss: 0.6658 - val_loss:
1.3905
Epoch 46/100
13/13 [==============================] - ETA: 0s - loss: 0.6300
Epoch 46: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 105ms/step - loss: 0.6300 - val_loss:
1.4332
Epoch 47/100
13/13 [==============================] - ETA: 0s - loss: 0.6101
Epoch 47: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 87ms/step - loss: 0.6101 - val_loss:
1.4121
Epoch 48/100
12/13 [===========================>...] - ETA: 0s - loss: 0.5703
Epoch 48: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 86ms/step - loss: 0.5802 - val_loss:
1.4566
Epoch 49/100
12/13 [===========================>...] - ETA: 0s - loss: 0.5809
Epoch 49: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 85ms/step - loss: 0.5903 - val_loss:
```

```
1.4495
Epoch 50/100
12/13 [==========================>...] - ETA: 0s - loss: 0.5405
Epoch 50: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 89ms/step - loss: 0.5445 - val_loss:
1.4037
Epoch 51/100
12/13 [==========================>...] - ETA: 0s - loss: 0.5836
Epoch 51: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 86ms/step - loss: 0.5862 - val_loss:
1.4154
Epoch 52/100
13/13 [==============================] - ETA: 0s - loss: 0.5423
Epoch 52: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 87ms/step - loss: 0.5423 - val_loss:
1.4402
Epoch 53/100
12/13 [==========================>...] - ETA: 0s - loss: 0.5330
Epoch 53: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 85ms/step - loss: 0.5416 - val_loss:
1.4383
Epoch 54/100
12/13 [==========================>...] - ETA: 0s - loss: 0.4960
Epoch 54: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 86ms/step - loss: 0.5170 - val_loss:
1.4466
Epoch 55/100
12/13 [==========================>...] - ETA: 0s - loss: 0.5068
Epoch 55: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 86ms/step - loss: 0.5157 - val_loss:
1.4218
Epoch 56/100
12/13 [==========================>...] - ETA: 0s - loss: 0.5196
Epoch 56: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 86ms/step - loss: 0.5192 - val_loss:
1.4600
Epoch 57/100
12/13 [==========================>...] - ETA: 0s - loss: 0.5097
Epoch 57: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 85ms/step - loss: 0.5029 - val_loss:
1.4392
Epoch 58/100
12/13 [==========================>...] - ETA: 0s - loss: 0.4825
Epoch 58: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 84ms/step - loss: 0.4907 - val_loss:
1.4693
Epoch 59/100
13/13 [==============================] - ETA: 0s - loss: 0.5050
Epoch 59: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 84ms/step - loss: 0.5050 - val_loss:
1.4294
Epoch 60/100
13/13 [==============================] - ETA: 0s - loss: 0.4871
Epoch 60: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 86ms/step - loss: 0.4871 - val_loss:
1.4807
Epoch 61/100
12/13 [==========================>...] - ETA: 0s - loss: 0.4493
Epoch 61: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 85ms/step - loss: 0.4512 - val_loss:
1.4571
```

```
Epoch 62/100
13/13 [==============================] - ETA: 0s - loss: 0.4154
Epoch 62: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 87ms/step - loss: 0.4154 - val_loss:
1.5155
Epoch 63/100
12/13 [==========================>...] - ETA: 0s - loss: 0.4283
Epoch 63: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 83ms/step - loss: 0.4234 - val_loss:
1.5083
Epoch 64/100
12/13 [==========================>...] - ETA: 0s - loss: 0.4418
Epoch 64: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 84ms/step - loss: 0.4332 - val_loss:
1.4855
Epoch 65/100
13/13 [==============================] - ETA: 0s - loss: 0.4177
Epoch 65: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 86ms/step - loss: 0.4177 - val_loss:
1.4770
Epoch 66/100
12/13 [==========================>...] - ETA: 0s - loss: 0.4229
Epoch 66: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 89ms/step - loss: 0.4279 - val_loss:
1.4948
Epoch 67/100
13/13 [==============================] - ETA: 0s - loss: 0.3804
Epoch 67: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 88ms/step - loss: 0.3804 - val_loss:
1.5133
Epoch 68/100
13/13 [==============================] - ETA: 0s - loss: 0.4119
Epoch 68: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 110ms/step - loss: 0.4119 - val_loss:
1.5374
Epoch 69/100
13/13 [==============================] - ETA: 0s - loss: 0.3911
Epoch 69: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 114ms/step - loss: 0.3911 - val_loss:
1.5201
Epoch 70/100
13/13 [==============================] - ETA: 0s - loss: 0.3939
Epoch 70: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 112ms/step - loss: 0.3939 - val_loss:
1.5009
Epoch 71/100
13/13 [==============================] - ETA: 0s - loss: 0.3473
Epoch 71: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 100ms/step - loss: 0.3473 - val_loss:
1.5528
Epoch 72/100
13/13 [==============================] - ETA: 0s - loss: 0.3433
Epoch 72: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 94ms/step - loss: 0.3433 - val_loss:
1.5478
Epoch 73/100
13/13 [==============================] - ETA: 0s - loss: 0.3531
Epoch 73: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 98ms/step - loss: 0.3531 - val_loss:
1.5533
Epoch 74/100
```

```
13/13 [==============================] - ETA: 0s - loss: 0.3421
Epoch 74: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 98ms/step - loss: 0.3421 - val_loss:
1.5683
Epoch 75/100
13/13 [==============================] - ETA: 0s - loss: 0.3050
Epoch 75: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 105ms/step - loss: 0.3050 - val_loss:
1.6099
Epoch 76/100
13/13 [==============================] - ETA: 0s - loss: 0.3080
Epoch 76: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 110ms/step - loss: 0.3080 - val_loss:
1.6318
Epoch 77/100
12/13 [==========================>...] - ETA: 0s - loss: 0.3084
Epoch 77: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 85ms/step - loss: 0.3068 - val_loss:
1.6066
Epoch 78/100
13/13 [==============================] - ETA: 0s - loss: 0.2940
Epoch 78: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 112ms/step - loss: 0.2940 - val_loss:
1.6228
Epoch 79/100
13/13 [==============================] - ETA: 0s - loss: 0.2920
Epoch 79: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 92ms/step - loss: 0.2920 - val_loss:
1.6695
Epoch 80/100
13/13 [==============================] - ETA: 0s - loss: 0.3089
Epoch 80: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 90ms/step - loss: 0.3089 - val_loss:
1.6563
Epoch 81/100
13/13 [==============================] - ETA: 0s - loss: 0.3075
Epoch 81: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 98ms/step - loss: 0.3075 - val_loss:
1.6199
Epoch 82/100
12/13 [==========================>...] - ETA: 0s - loss: 0.2917
Epoch 82: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 94ms/step - loss: 0.2940 - val_loss:
1.6492
Epoch 83/100
13/13 [==============================] - ETA: 0s - loss: 0.2840
Epoch 83: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 92ms/step - loss: 0.2840 - val_loss:
1.6746
Epoch 84/100
13/13 [==============================] - ETA: 0s - loss: 0.2850
Epoch 84: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 94ms/step - loss: 0.2850 - val_loss:
1.6943
Epoch 85/100
13/13 [==============================] - ETA: 0s - loss: 0.2888
Epoch 85: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 85ms/step - loss: 0.2888 - val_loss:
1.7056
Epoch 86/100
13/13 [==============================] - ETA: 0s - loss: 0.3007
```

```
Epoch 86: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 93ms/step - loss: 0.3007 - val_loss:
1.7055
Epoch 87/100
13/13 [==============================] - ETA: 0s - loss: 0.2751
Epoch 87: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 93ms/step - loss: 0.2751 - val_loss:
1.6637
Epoch 88/100
12/13 [==========================>...] - ETA: 0s - loss: 0.2815
Epoch 88: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 84ms/step - loss: 0.2856 - val_loss:
1.6806
Epoch 89/100
12/13 [==========================>...] - ETA: 0s - loss: 0.2870
Epoch 89: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 93ms/step - loss: 0.2839 - val_loss:
1.7298
Epoch 90/100
12/13 [==========================>...] - ETA: 0s - loss: 0.2539
Epoch 90: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 88ms/step - loss: 0.2602 - val_loss:
1.6860
Epoch 91/100
13/13 [==============================] - ETA: 0s - loss: 0.2501
Epoch 91: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 91ms/step - loss: 0.2501 - val_loss:
1.6805
Epoch 92/100
13/13 [==============================] - ETA: 0s - loss: 0.2409
Epoch 92: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 90ms/step - loss: 0.2409 - val_loss:
1.6914
Epoch 93/100
13/13 [==============================] - ETA: 0s - loss: 0.2484
Epoch 93: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 106ms/step - loss: 0.2484 - val_loss:
1.7194
Epoch 94/100
12/13 [==========================>...] - ETA: 0s - loss: 0.2299
Epoch 94: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 109ms/step - loss: 0.2364 - val_loss:
1.6860
Epoch 95/100
13/13 [==============================] - ETA: 0s - loss: 0.2265
Epoch 95: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 86ms/step - loss: 0.2265 - val_loss:
1.7675
Epoch 96/100
13/13 [==============================] - ETA: 0s - loss: 0.2232
Epoch 96: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 91ms/step - loss: 0.2232 - val_loss:
1.7801
Epoch 97/100
12/13 [==========================>...] - ETA: 0s - loss: 0.2245
Epoch 97: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 89ms/step - loss: 0.2295 - val_loss:
1.8160
Epoch 98/100
13/13 [==============================] - ETA: 0s - loss: 0.2251
Epoch 98: val_loss did not improve from 1.39053
```

```
13/13 [==============================] - 1s 90ms/step - loss: 0.2251 - val_loss:
1.8040
Epoch 99/100
13/13 [==============================] - ETA: 0s - loss: 0.2613
Epoch 99: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 91ms/step - loss: 0.2613 - val_loss:
1.7942
Epoch 100/100
13/13 [==============================] - ETA: 0s - loss: 0.2612
Epoch 100: val_loss did not improve from 1.39053
13/13 [==============================] - 1s 90ms/step - loss: 0.2612 - val_loss:
1.7675
```

In [37]:
```
#Loading best model
from keras.models import load_model
model = load_model('best_model.h5')
```

In [35]:
```
import random
ind = np.random.randint(0,len(x_val)-1)

random_music = x_val[ind]

predictions=[]
for i in range(20):

    random_music = random_music.reshape(1,no_of_timesteps)

    prob  = model.predict(random_music)[0]
    y_pred= np.argmax(prob,axis=0)
    predictions.append(y_pred)

    random_music = np.insert(random_music[0],len(random_music[0]),y_pred)
    random_music = random_music[1:]

print(predictions)
```

```
1/1 [==============================] - 0s 308ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 39ms/step
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 49ms/step
1/1 [==============================] - 0s 37ms/step
1/1 [==============================] - 0s 38ms/step
1/1 [==============================] - 0s 40ms/step
1/1 [==============================] - 0s 44ms/step
1/1 [==============================] - 0s 41ms/step
1/1 [==============================] - 0s 33ms/step
1/1 [==============================] - 0s 36ms/step
1/1 [==============================] - 0s 48ms/step
1/1 [==============================] - 0s 50ms/step
1/1 [==============================] - 0s 43ms/step
1/1 [==============================] - 0s 160ms/step
1/1 [==============================] - 0s 69ms/step
1/1 [==============================] - 0s 52ms/step
1/1 [==============================] - 0s 54ms/step
[15, 38, 23, 38, 15, 3, 15, 3, 15, 3, 3, 3, 3, 3, 15, 23, 23, 23, 23, 15]
```

In [38]:
```
x_int_to_note = dict((number, note_) for number, note_ in enumerate(unique_x))
predicted_notes = [x_int_to_note[i] for i in predictions]
```

In [39]:
```
def convert_to_midi(prediction_output):

    offset = 0
    output_notes = []

    # create note and chord objects based on the values generated by the model
    for pattern in prediction_output:

        # pattern is a chord
        if ('.' in pattern) or pattern.isdigit():
            notes_in_chord = pattern.split('.')
            notes = []
            for current_note in notes_in_chord:

                cn=int(current_note)
                new_note = note.Note(cn)
                new_note.storedInstrument = instrument.Tambourine()
                notes.append(new_note)

            new_chord = chord.Chord(notes)
            new_chord.offset = offset
            output_notes.append(new_chord)

        # pattern is a note
        else:

            new_note = note.Note(pattern)
            new_note.offset = offset
            new_note.storedInstrument = instrument.Tambourine()
            output_notes.append(new_note)

        # increase offset each iteration so that notes do not stack
        offset += 1
    midi_stream = stream.Stream(output_notes)
    midi_stream.write('midi', fp='music.mid')
```

In [41]:
```
convert_to_midi(predicted_notes)
```

In [ ]:

In [ ]: