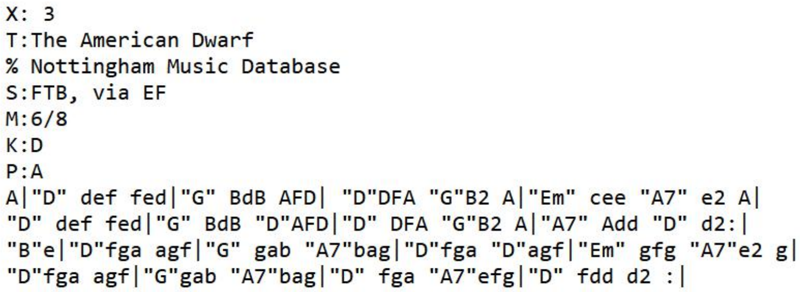
Music Generation Using RNN

In this section, we use Recurrent Neural Network(RNN) to generate music automatically. The first step of our task is training a RNN model using music datasets. Once the model learns some features / characters of music, we can use it to produce some new music.

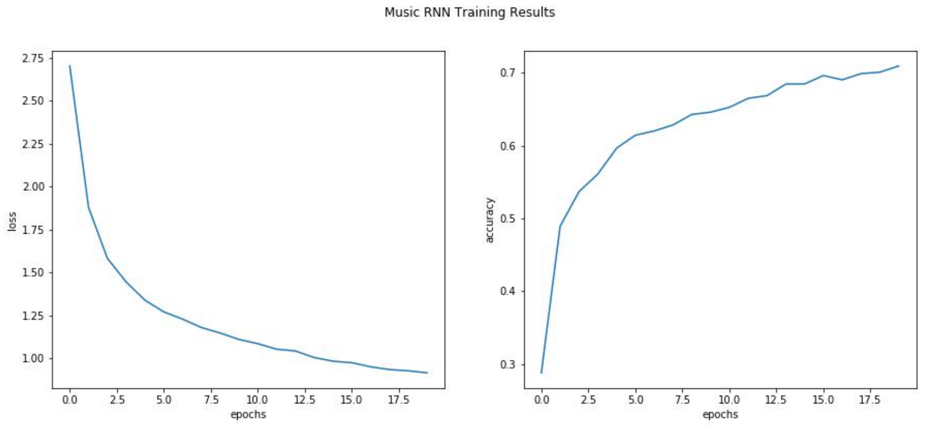
The dataset we use is the ABC version of the Nottingham Music Database, the ABC notation is a form of musical notation for both human and computers. The following figure show a piece of music represented by ABC notation.



```
X: 3
T:The American Dwarf
% Nottingham Music Database
S:FTB, via EF
M:6/8
K:D
P:A
A|"D" def fed|"G" BdB AFD| "D"DFA "G"B2 A|"Em" cee "A7" e2 A|
"D" def fed|"G" BdB "D"AFD|"D" DFA "G"B2 A|"A7" Add "D" d2:|
"B"e|"D"fga agf|"G" gab "A7"bag|"D"fga "D"agf|"Em" gfg "A7"e2 g|
"D"fga agf|"G"gab "A7"bag|"D" fga "A7"efg|"D" fdd d2 :|
```

As we can see, the first 7 lines records some meta data of the music, the rest part represents the tune. The tune in ABC notation is constituted of a sequence of characters which allow us treat it as sentences.

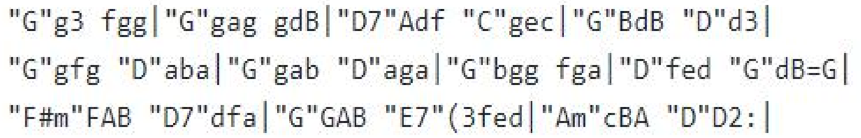
Given the sequential nature of music and the power of ABC notation of representing music as characters, we train a RNN and use it to produce new music. To be more specific, we need to use a Many-to-Many RNN, where the input and output have the same dimensionality. This is because we feed the character of input sequences one by one to the model, and we set the next character as the output / target of the current character. By doing so, we hope the model learns how to predict next character given the current character. We use one embedding layer to wrap the input and then two GRU layers and one dense layer to build our model, we also add two dropout layers to prevent overfitting. The following figure shows our training results.



According to the results, the accuracy achieves above 70% after 20 epochs. The training loss also decrease accordingly. Thus, we can assume that the RNN model has understood music to some extent. Thus, in the next step, we use the trained model to generated new music.

As for music generation, we need to specify an initial character and the length of sequence we want the music to have. We sample both the initial character and the length. And then we feed the initial character to our RNN, the output will be the next input. We repeat the process until we reach the target length.

The following figure shows the generated music after some clean up, i.e. we remove the meta data and some meaningless characters.



```
"G"g3 fgg|"G"gag gdB|"D7"Adf "C"gec|"G"BdB "D"d3|
"G"gfg "D"aba|"G"gab "D"aga|"G"bgg fga|"D"fed "G"dB=G|
"F#m"FAB "D7"dfa|"G"GAB "E7"(3fed|"Am"cBA "D"D2:|
```

Now, based on our observation, the generated music resemble the sample music we illustrated before to some extent. This proves our RNN model has learned some pattern / knowledge of music and is able to generated new music based on knowledge it learned.