Report and Important Information

A musical sequence can be of two types -

- Monophony - Only one note can be played at one time
- Polyphony - Multiple notes can be played at one time

Types of representations used to represent music files-

- MIDI Representation - Symbolic representation of a music file used by Magenta. Symbolic data represents music in terms of composition and harmony. This MIDI format is converted into and from a Protocol Buffer called 'NoteSequence'

- Audio Waveform Representation - Graph displaying aptitude changes over time. Amplitude of the waveform is represented by decibel (dB) and pitch is represented by hertz (Hz). Earlier, using such a raw waveform was uncommon as higher computational power was needed. Recent advances make it at par with other representations.

- Spectrogram Representation - Spectrogram is a result of doing Fourier transform on the audio stream. It decomposes the audio signal giving us the intensity of a frequency band (i.e small split of whole segment). Spectrogram lets you actually see the content of the music.

Definition : Task of music generation - Iteratively generate a new sequence by predicting the next notes from an input sequence (priming track) at each iteration. (analogous to music score prediction network)

RNN is used for music generation as it works on a sequence of vectors, and thus the i/p and o/p sizes can be arbitrary values.
RNN remembers the past to predict the future better, i.e., o/p of every state vector is used to update the state vector for the next step. At each step the RNN uses the hidden vector and input vector to make a prediction. Hence, RNN prediction is based on its recurrent connection, which keeps track of the context, and doesn't rely on the input alone. LSTM mechanism helps in keeping track of how far the gradients can roll out during backpropagation. In Magenta, all RNNs use LSTM cells.

Magenta music generation in TensorFlow -

Magenta is distributed as an open source Python library, powered byTensorFlow. This library includes utilities for manipulating source data (primarily music and images), using this data to train machine learning models, and finally generating new content from these models.

Models are stored as bundle files which contain the checkpoints and the metadata. The pre-trained model contains the weights from the training phase that are used to initialize the RNN network. A bundle file is a file containing the model checkpoint and metadata.

To generate a drum sequence - we download the bundle file, drum_kit_rnn.mag, in the bundles folder. Magenta encodes primer sequence for the model to understand, and uses it to initialize the model. It loops until all the steps have been generated. It calculates the negative log-likelihood of the resulting sequence, and updates the model state for the next iteration. In the end, it prunes the generated branches using beam search. We generate MIDI files in the output directory.

Important parameters of the model that can be tweaked -

- *num_steps* : Total number of steps in the generated drum track. Here, total steps = priming drum track length + generated steps.

- *qpm* : used to change the tempo (or speed) at which the track is played

- *primer_drums* : A string representation of a list of tuples containing drum pitch values, used to pass self defined priming drum track

- *primer_midi* : MIDI file containing drum track that will be used as primer

- *temperature* : Changes the randomness of the generated track. Greater than 1.0 makes tracks more random, less than 1 makes them less random.

Important calculations in musical sequences -

1 bar = 16 steps
Tempo = Quarter-notes per minute (qpm) = no. of seconds required for 1 bar to play
Duration of a sequence = ((60/qpm)/4) * num_steps