Automated Music Generation using LSTM

In this work , it was tried to produce single-instrument music using LSTM, which is a known model of RNN . The training data used are music files in MIDI format and containing a single instrument . Basically, the reason for using LSTM is that as the network learns to create music and the data is sequential data, it provides avoid from long term dependency problem . In addition, LSTM cells has the ability to merge and delete data that it contains as a model . Music21 library was used to make the datasets applicable to the model . With the help of Music21, the dataset was transformed into objects in the form of notes and chords . Notes have offset, pitch values, and octave. Chords have set of notes. Pitch value is defined as the degree of sound depending on the frequency of the sound . Octave is defined as a interval of one pitch. For example, from G4 to G5. The pitch values ​​are added as strings to the Music21 stream object that contains the notes and chords. All notes (including the contents of the chords) are separated from each other by dot and encoded into a single line string. Therefore output of such network can be easily decoded. By using the mapping function, string based data made integer based. This is done because machine learning works better than string based ML. Technical processes have been made to enable the model to learn better (like reducing overfitting) by using Dropout layers, dense layers and activation layers . The beginning point is selected randomly from the list but in any case if one wants to control the starting point then a function can be created to replace the current random function. They can determine the number of notes contained in the music we will produce in this work . In this way , they had the chance to change the duration of the music produced.

Continuous recurrent neural networks with adversarial training

Different from above two studies, GAN was used in this study. The approach in GAN is as follows: it involves training two neural models with conflicting goals , a generator (G) and a discriminator (D), which forces each other to improve . The generator tries to produce samples that look real and the discriminator tries to distinguish between produced samples and real data . The result is a zero-sum game, which is called the Nash equilibrium. The music data used in this study are in MIDI format. These are examples of classical music produced by different composers. Tone length, frequency, intensity and time spent since the previous tone values ​​are used to model this data . This allows us to represent polyphonic chords. Each midi event of the type “note on” was loaded and saved together with its duration, tone, intensity (velocity), and time since beginning of last tone. The tone data is internally represented with the corresponding sound frequency. Internally, all data is normalized to a tick resolution of 384 per quarter note. Backpropagation through time (BPTT) and mini-batch stochastic gradient descent was used. Learning rate was set to 0.1, and we apply L2 regularization to the weights both in G and D. If D is stronger than G, say, 70% for loss G, stop updating D. Feature matching avoid overfitting, the manufacturer produces more realistic data, making the internal structure of D to match the real data.

STYLE-CONDITIONED MUSIC GENERATION

In this work, they proposed a formulation to the VAE that allows users to condition on the style of the generated music. In variational autoencoders, the input passed through the encoder is encoded as a probability distribution. If we consider this probability distribution as a normal Distribution, we need mean and variance values ​​to express this distribution. In other words, encoder output in variational autoencoders is not a code, but mean and variance values. The code is obtained by sampling from these mean and variance values. This code can be decoded with decoder.To train the model, a technique called the reparametrization trick is used because the code sampling process from mean and variance values ​​is not suitable for backpropagation. A value of is used to train the VAE using backpropagation. The loss function is calculated as (Reconstruction loss + Kullback-Leibler divergence). But in addition to this, the styles of the music are also discussed in this study. That's why latent space is divided into two. In second part of latent space gumbel distribution trick was used because it was difficult to get the gradients of discrete data which are music styles. Thus, styles of the music are kept embedded in the style codebook. So network learns styles also. JSB and NMD datasets are used as data. The JSB compositions written in four parts: soprano, alto, tenor and bass. In contrast, the folk tunes in the NMD consist of a simple monophonic melody line, accompanied by a chord sequence. Data is represented with the help of pretty-midi library. Also, it was aimed that the proposed model would predict better style of music by transferring style on the test set.