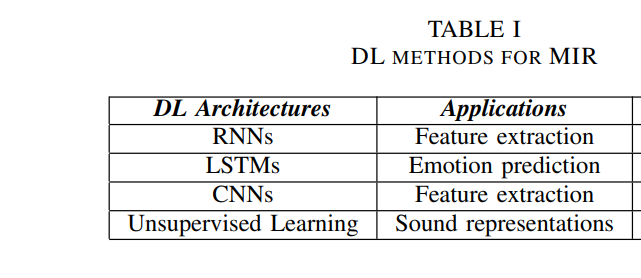
Music Information Retrieval (MIR) MIR refers to the extraction of useful information from music data.

The DL architectures that are most frequently employed for MIR tasks are: i) Recurrent Neural Networks (RNNs), and ii) Convolutional Neural Networks (CNNs).

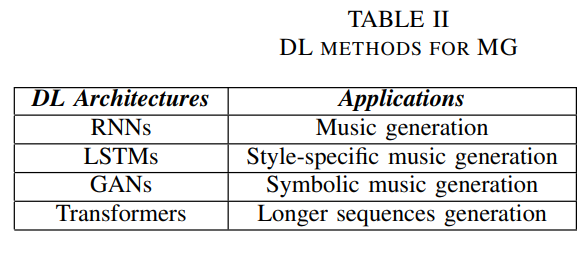


RNNs are a family of neural networks for processing sequential data [1]. A subset of RNNs which has been successfully applied in many different areas including MIR is the Long Short Term Memory networks (LSTM)

CNNs are a class of DL models that are capable of processing data with a known grid-like topology [1]. CNNs make use of the convolution operation instead of matrix multiplication in at least one of their layers [1]. CNNs are incredibly successful in numerous tasks such as CV, NLP, time series forecasting etc [15]. In the field of MSP and especially MIR, working with time - frequency data, CNNs are frequently employed, in order to extract local information from music data.

Overfitting is an always present issue in DL. In order to avoid it, data augmentation may be employed. This approach was followed in [29] for the task of the separation of music into individual instrument tracks. In addition a combination of a feed-forward neural network with a RNN performed better than the individual models themselves.

DL-based MG makes use of the results that are produced by MIR methods. The most common approaches to MG are: i) RNNs - LSTMs, ii) Generative Adversarial Networks (GANs), and iii) Transformers



Chords play a crucial role in music composition, so the task of chord generation is an important one. In [36], [41] bi-directional LSTMs are used for this problem

GANs were first introduced in [43]. The core idea behind GANs is the existence of two antagonistic entities; the generator and the discriminator. Given a training set of real samples, the generator is trained to approximate the real data distribution, while the discriminator tries to discriminate between real and synthetic samples.

The core idea behind transformers is the mechanism of self-attention, which refers to the process of differentially weighting the significance of each part of the input data. Transformers are designed to handle sequential input data, but they do not necessarily process the data in order.

great concern is the computational cost of the DL models’ training. Reducing the required memory and producing longer sequences of music data will result in many more commercial applications, testing in this way the models to the real world’s necessities

**Second: LSTMs**

Neural networks like Long-Short Term Memory (LSTM) learn the data from the prepared music file, and produce a brand new one for the users, which helps them understand the styles and structures of music. Experiments proves the best group of parameters for the designed LSTM neutral network to make the music more natural and smoother.

For one part, the neural network LSTM is used, establishing a path to learn from the pattern of the midi files the users provide. While a package named music21 is inserted to preprocess the template midi files. The users will only need to pick a midi file of the music types they like for the system to learn and operate. The operations are simple enough, and the deviations are also in a controllable range. For the other one, notes and chords are provided in different classes through the music21 package.

It's always a big barrier for RNN to remember long term information, while LSTM successfully overcomes such problem by detailing its repeating modules. Both of which have a tanh layer, but LSTM contains four neural network layers instead of just one in RNN, interacting in a more profound way.

Different parts in LSTM module, cell state is the most important. Other parts in structure are called gates which regulate interactions with the information. There are activation functions embedded in hidden states transmitting the linear form of information in the previous states and the inputs.

Sigmoid func – most frequently used. Relatively easier to take derivative. Helps filter information and regulation process is more concise. Gradient Vanishing(Problem) makes derivatives approach 0, hinders update of parameters. Sacrifices of rate of convergence. The loss will become greater, and the accuracy of its learning process will also be affected.

Tanh func – Solves unicity problem, but vanishing gradient still present.

ReLU func: solves the problem of gradient vanishing, and holds a high learning rate. Moreover, its convergence rate outweighs both Sigmoid and tanh function. Problem here is activation process can fail.

Leaky ReLU: order to solve the problem of failing in activation processes, this function is proposed. It changes the slope of the part less than zero into 0.01.

Losses: MSE, Loss Entropy,

Optimizer: Stochastic Gradient Descent, AdaGrad(requires global learning rate, which helps automate learning rate, its flexible for more situations)

Their Model

The frame of our model is consists of six layers. The first two layers are LSTMs, which provide a sequence for the input. They help the dense layers to familiarize with the information of the data sets. The dense layers then begin their learning process of those data sets, and will provide an output for the following updating process.