Generating Music with LSTM

By Aleksander Sørup Lund (s153827)
and Emma Silke Borre (s162590)

DTU
Course – Deep Learning (02456)
Supervisor – Valentin Liévin

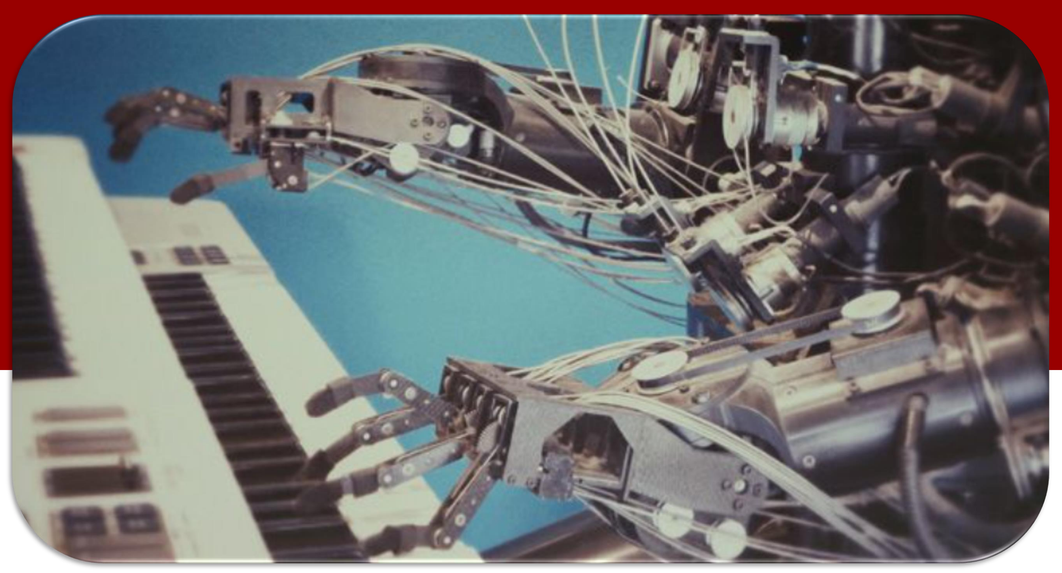

Fig 1. Source [1]

Problem statement

The aim in this project is to create a deep neural network capable of creating new and interesting music in the style of the biggest classical music composers such as Bach, Liszt and Chopin. The data used for the project is the Maestro 2.0.0 data set, which contains midi file encoded music pieces corresponding to more than 200 hours of music. The set only contains piano recordings, which reduces the complexity of the data. [2]. Long Short Term Memory (LSTM) is a recurrent neural network, with the specific purpose of being able to contain long term memory. As music by nature inherits complex temporal patterns, LSTM would be ideal for creating a generative model for music.

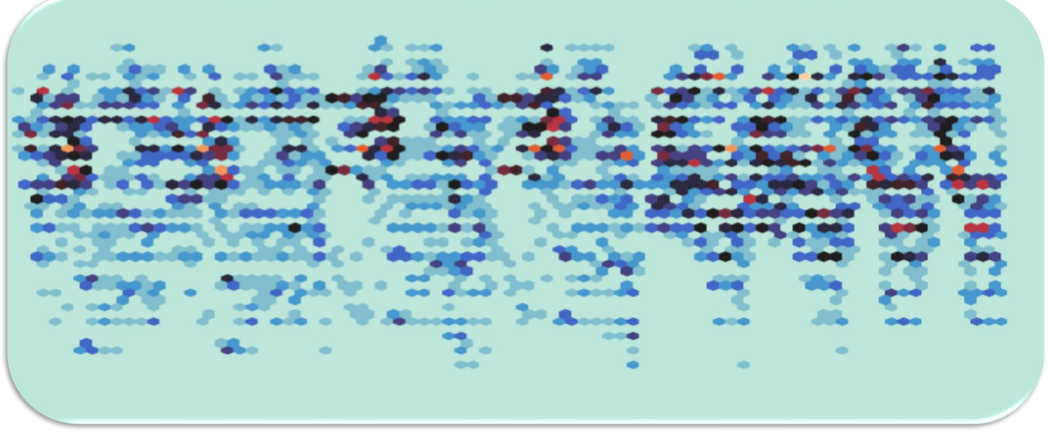

Fig 2. Example of midi training data, with notes vertically and timing horisontally

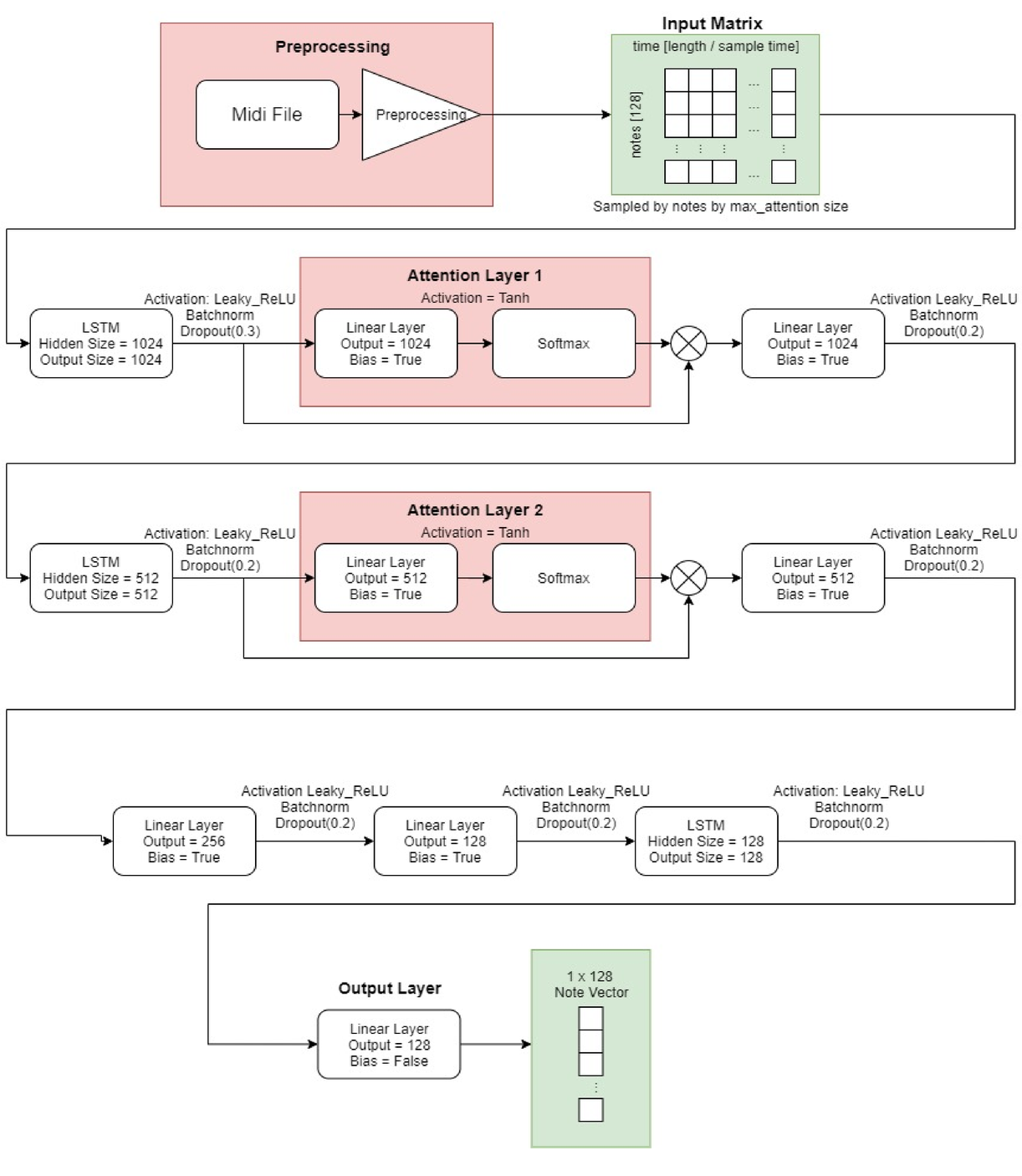

Fig 3. Deep Learning Architecture and program flow chart

Midi Data extraction

In order to extract information from the midi data, a library called MIDO is used to store the musical pieces as a series of events, with information of notes and timing. These events are sampled with an invariant frequency and represented in a matrix with a note- and time dimension. A midi file is in general described with beats, where each beat is four events. For a piece that is played with 120 BPM, the sampling frequency will be 20 Hz or 0.05 s. The dataset contains pieces of varying length, and is therefore truncated with a limit of seven minutes, whereafter the pieces that are shorter are zero-padded to achieve the same length in all pieces.

Deep Learning architecture and training

Complex patterns require complex deep networks. Looking at previous work from various sources, [3], a deep learning architecture is proposed. The LSTM networks with a high sense of harmony and patterns, contained attention cells. These layers are used to calculate the importance of previous notes in the input sequence, by multiplying the attention and the LSTM output, where the softmax thus scales each states importance for the following LSTM and Linear layers. Using multiple of these attention layers seemed to help performance and keep the network output harmonious sequences for longer periods. For training, a SGD optimizer and Binary Cross Entropy Loss was used. To calculate the loss, the output of the neural network was compared with the next timestep. Learning rates are varied through testing with an optimal range was between 0.001 and 0.01.

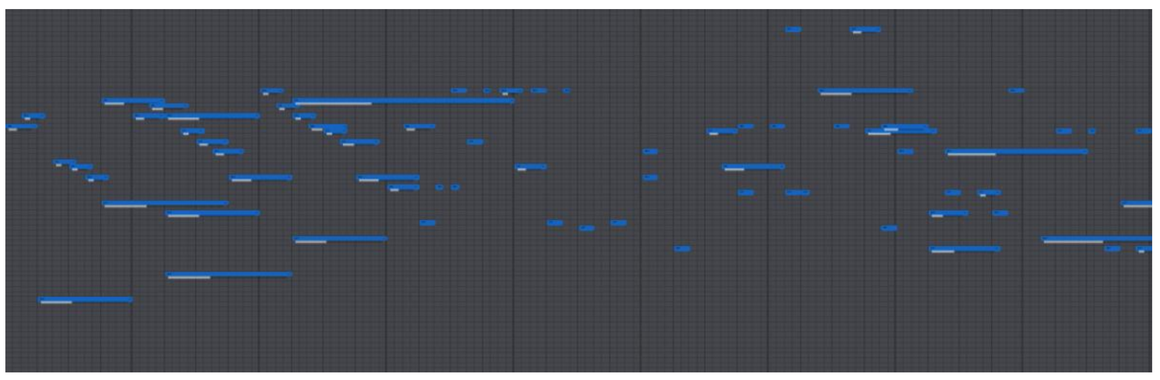

Fig 4. Generated midi file from the network, primed with the first eight beats

References
[1] https://www.designguide.dtu.dk/standard-brand-assets?fbclid=IwAR0Xa3P1yiFWh3DdRUR2_X2-a0rnB_eiq3nOvJJ3gX6C2mPT3CtwpXHKxNU#standard-user-printed-materials-banner
[2] https://magenta.tensorflow.org/datasets/maestro
[3] https://hedonistrh.github.io/2018-04-27-Music-Generation-with-LSTM/?fbclid=IwAR3pNtFVnyNlsQAkCsrul9ZCxmyk9R5GXeGeGSb6bvBDRvQt4ZSpLC96V68
[4] https://www.dropbox.com/sh/0o8ttmtlr7kxc20/AAD4WARAK5Y9Y5n3KjU91jgqa?dl=0&fbclid=IwAR1wIIM6s4-Rk3OE4yrBn-JqP7S-l3PL-RwQI5BvqX2X_g_sCNm0IDMg630