# LSTM for Piano Music Generation


**Anadon H., Bou A., Khan T., Myriokefalitakis P.**

DT2119 - Project Group 13



**Abstract**

In this project we try to generate piano music by using a Long Short-Term Memory (LSTM). More precisely, we feed the network with a dataset of piano music in the midi file format, following the note-by-note approach. Our aim is to get a sound comparable to the original one from the dataset. The reason of choosing this architecture stems from the ability of these networks to remember states from the past, a property that is especially important when working with time sequences.


## Introduction

The aim of this project was to generate piano music given a data set of piano songs in the midi format using a RNN (Recurrent Neural Network) architecture namely LSTM (Long short-term memory).

+ RNN has an advantage over DNN: Ability to maintain internal memory

x Problem of vanishing gradients.

**Solution:** LSTM $\Longrightarrow$ maintain a long-term and a short-term memory.

## Approach:

- Note-by-note approach
- Not raw audio approach (computationally complex)
- Keras on top of TensorFlow
- Dataset in midi format
- Music21 (aka digital alphabet for music, Python tool for note extraction)
- PDC: Tesla K80 nodes

## Method

### Data Representation and Feature Extraction

Data: files in Midi format (Musical Instrument Digital Interface).


+ very small size

+ all feature of the sound (pitch, velocity, volume etc.) can be edited

+ No interference or background noise in the data.

### Long short-term memory architecture

- RNN: Networks specialized in processing sequential data
  - input $\Longrightarrow$ a state
  - output $\Longrightarrow$ the following state
- LSTM: follows a RNN architecture. Advantage: Introduce a **memory cell** which permit to store old information to better capture long term dependencies
- Model Architecture and settings:
  - number of hidden layers: 2
  - number of nodes/hidden layer: 128
  - activation function: hyperbolic tangent
  - dropout(after every layer)
  - output layer: fully connected, softmax activation function

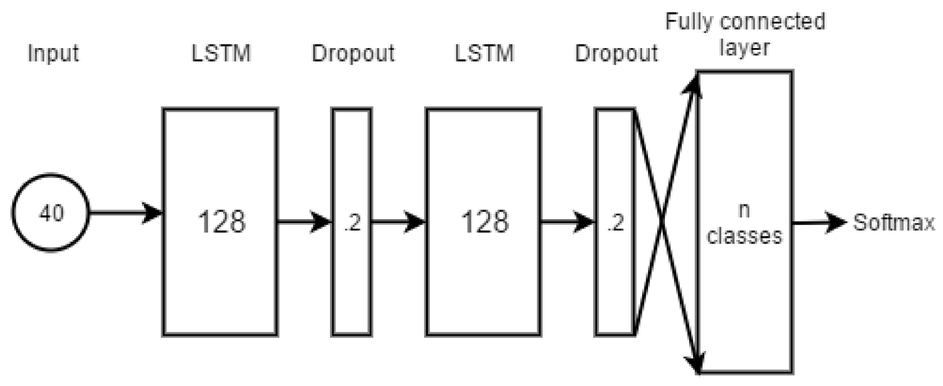


**Figure 1:** Model Architecture

- Training:
  - inputs: sequence of 40 states (with 3 state shift step)
  - labels: states that follows the last state of the window
  - loss function: categorical cross-entropy
  - update method: RMSProp
  - hyper-parameters: learning rate "$\eta$" (selection through cross validation) with momentum and without decay.
  - length of sequence to generate: 40
  - output/generation: set probabilities for each possible unique state that appear in the dataset.
- Generation of songs given a random seed and sampled with diversity (wider variety)

## Experiments

- Data:
  - 150 piano songs from several artists and from different time periods [1]
  - Only one instrument, piano
  - Features extracted: chords, notes, measures (key and time signature)
- Experiments:
  - Generate songs similar to the original ones from the dataset
- Hyper-parameter ($\eta$ and number of nodes/layer): Coarse to fine search. The best settings were chosen w.r.t the loss function

## Results

The loss function after training for 50 epochs can be seen in Fig. 2 and the generated pentagram of one song in Fig. 3. Further training required. In Fig. 4 we represent the songs using t-SNE to show that the generated songs (with different diversities) are similar to the training dataset.

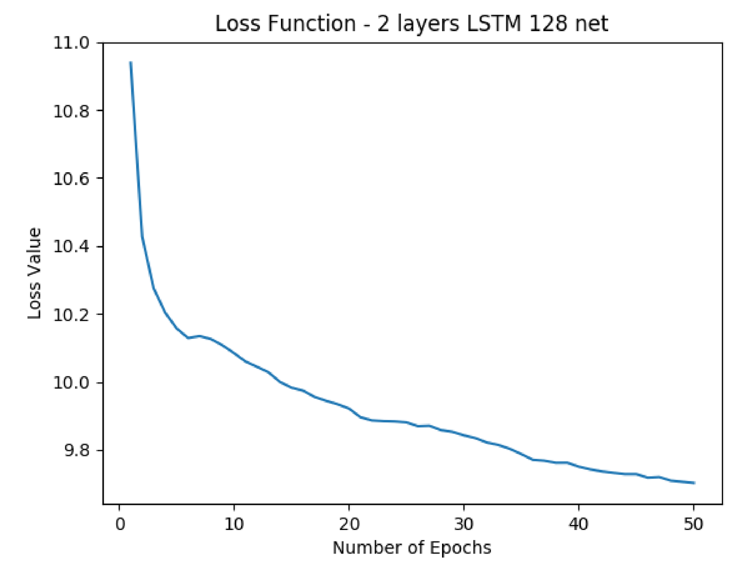


**Figure 2:** Example: Loss function after training with 150 songs for 50 epochs.

**Figure 3:** Pentagram with the first 10 measures of the song generated after training with 150 songs for 50 epoch.

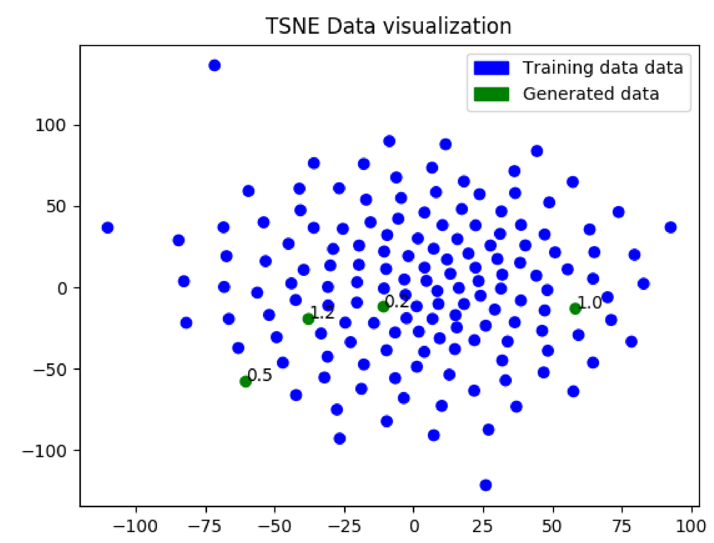


**Figure 4:** t-SNE songs representation resemblance

## Conclusions

- Longer training required as the lost function is still decreasing.
- Eta is tuned to have an stable learning rate
- The LSTM architecture can learn the musical patterns associated with time due to its ability to retain a memory
- **Future improvements:** Use all the properties midi uses in encoding a song into a a vector in order to model all the complexities present within a song like silences.

## References


[1] Classical piano midi page.

[2] Frank Brinkkemper. Analyzing 6 deep learning tools for music generation, October 2016.

[3] D. Eck and J. Laplme. Learning musical structure directly from sequences of music. *In Artificial Neural NetworksICANN*, 2008.

[4] Douglas Eck and Jurgen Schmidhuber. Learning the long-term structure of the blues. *In Artificial Neural NetworksICANN*, page 284289, 2002.

[5] Allen Huang and Raymond Wu. Deep learning for music. 2016.

[6] K. Cho J. Chung, C. Gulcehre and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *NIPS 2014 Deep Learning and Representation Learning Workshop*, 2014.

[7] Jan Koutnik Bas R. Steunebrink Jurgen Schmidhuber Klaus Greff, Rupesh Kumar Srivastava. Lstm: A search space odyssey. *arXiv:1503.04069v1*, 2015.

[8] Elkan Lipton, Berkowitz. A critical review of recurrent neural networks for sequence learning. *arXiv:1506.00019v4*, 2015.

[9] Michael C. Mozer. Neural network music composition by prediction: Exploring the benefits of psychoacoustic constraints and multi-scale processing. *Connection Science*, 5(6):247280, 1994.

[10] Vitelli Nayebi. Gruv: Algorithmic music generation using recurrent neural networks. 2016.

[11] Geoffrey Hinton with Nitish Srivastava Kevin Swersky. Lecture 6a, overview of mini-batch gradient descent. 2016.