Deep Learning Project

D7047E, Advanced Deep Learning

Luleå Tekniska Universitet

Fredrik Reinholdsen

Kalle Kalliorinne

Rasmus Hallén

Fabian de Laval

Github repo: https://github.com/Programermus/DeepDIVA

Scope of presentation

- Introduction
- Goals
- What others have done
- Datasets
    - Maestro (piano music)
    - Lakhnes (video game music)
- Models
- Implementation
    - PyTorch
    - Model structure
    - Two GRU (maestro dataset)
    - RNN (lakhnes dataset)
- Results
- Learning outcomes/conclusion

Introduction

Generating music with DL? How? Why?

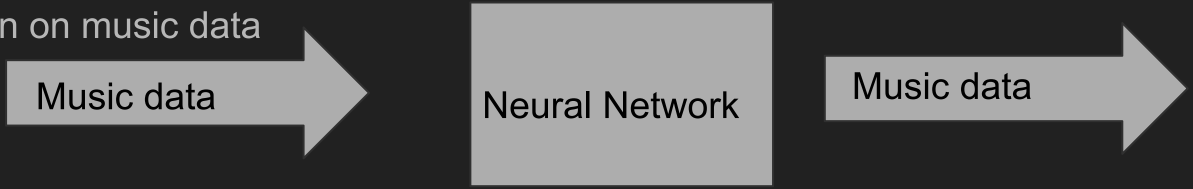
- Explore Artificial creativity
- Most people listen to music

Music is similar to language

- Context
- Rhythm
- Predictable

Train on music data

To generate music data



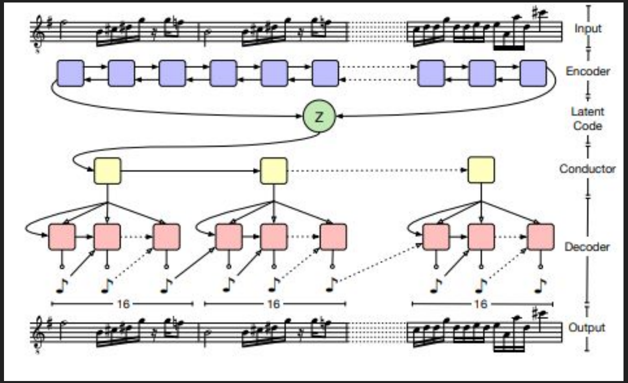
Goals of this project

- Make (Good) music with AI
- Learn pyTorch
- Implement RNNs in pyTorch
- Loading and pre-process datasets
- Construct a baseline for experiments
- Post-process output to appropriate file-format

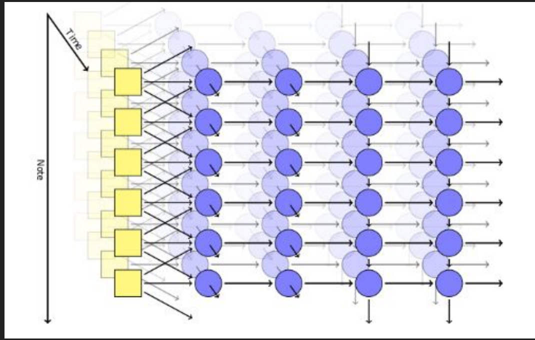
What others have done

- Magenta
  - Variational Autoencoders with RNNs
- openAI
  - Jukebox
  - Vector Quantization VAE   (VQ-VAE)
- Daniel Johnson
  - Biaxial LSTM
- LakhNES
  - https://chrisdonahue.com/LakhNES/

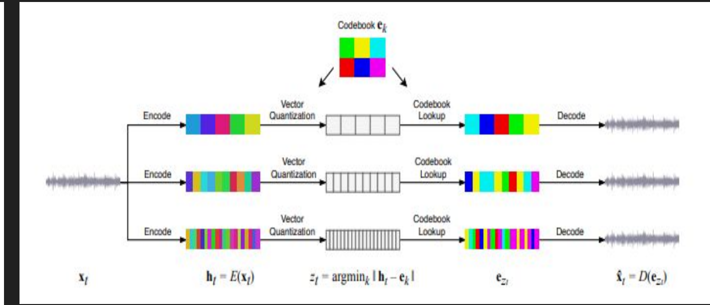
- “a first look at music composition using lstm recurrent neural networks” J. Schmidhuber 2002
- Many more...



Magenta paper
Source: https://arxiv.org/abs/1803.05428



Biaxial LSTM
Source: http://www.hexahedria.com/2015/08/03/composing-music-with-recurrent-neural-networks/



Jukebox paper
Source: https://arxiv.org/abs/2005.00341

MIDI - Datasets

- NES-MDB

  https://github.com/chrisdonahue/LakhNES
- Maestro

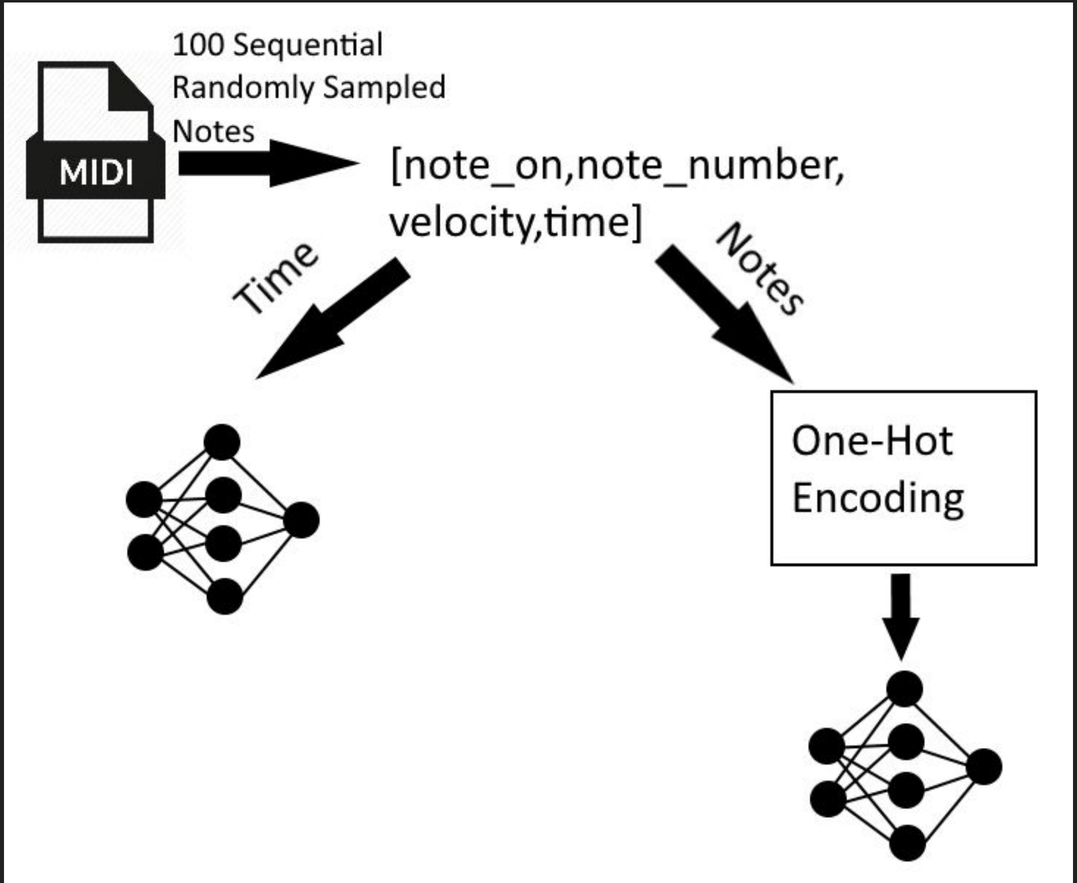
  https://magenta.tensorflow.org/datasets/maestro

Midi message: [track = 0, type = note_on, note = 64, velocity = 100, time = 384]

Implementation

MAESTRO:

- 2 separate Neural networks

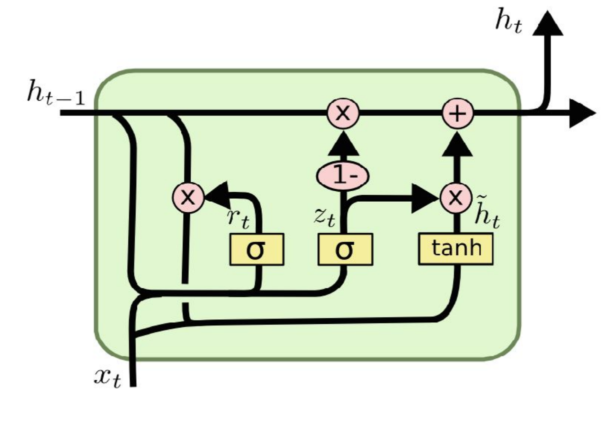


Model Structure MAESTRO

Note Input (batch_size, seq_len, 216)

- GRU Layer (256)
- Dropout Layer (0.3)
- GRU Layer (128)
- Dropout Layer (0.3)
- FC Layer (128)
- Dropout Layer (0.3)
- FC Layer (Input_size)

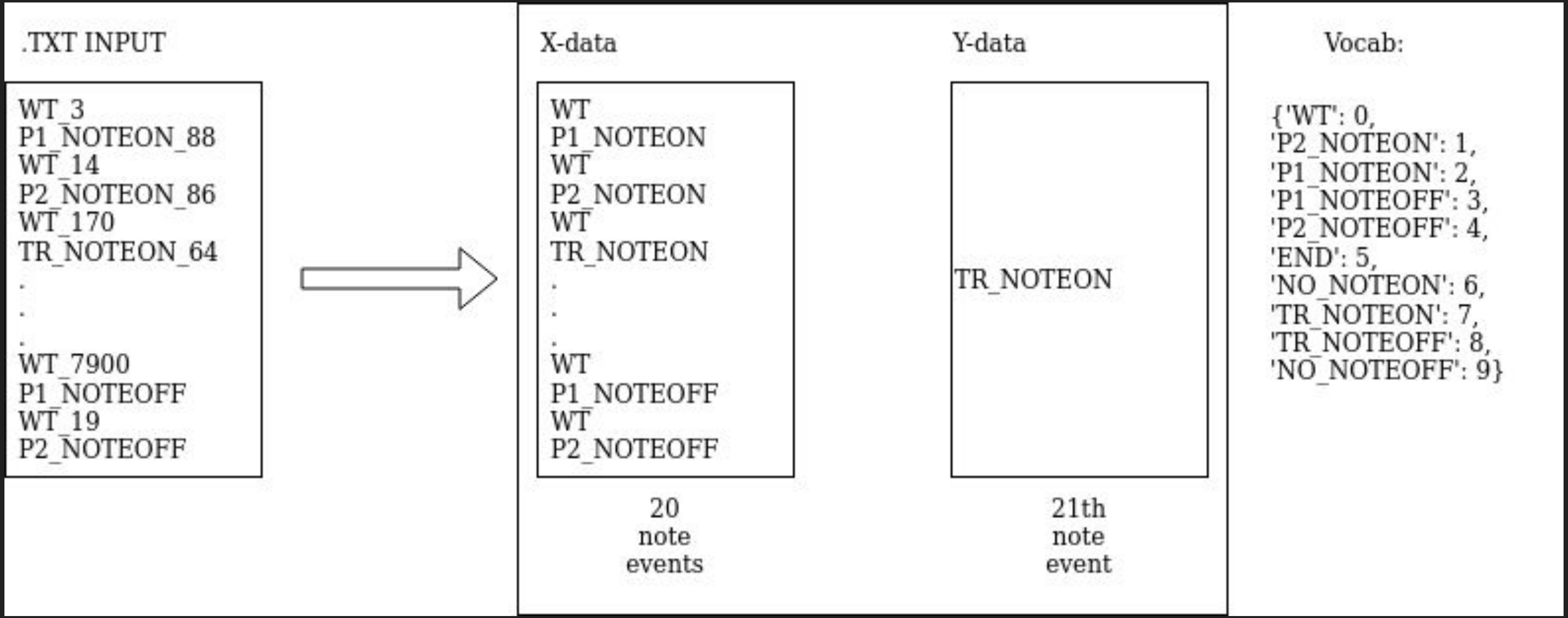
Time Input (batch_size, seq_len, 1)



GRU layer, Source:https://colah.github.io/posts/2015-08-Understanding-LSTMs/

RMSprop optimizer. MSE loss function for the time net, and CCE for the note net

Implementation Lakhnes



Structure

Model 1

- LSTM (64)
- Dropout (0.2)
- FC Layer

Model 2

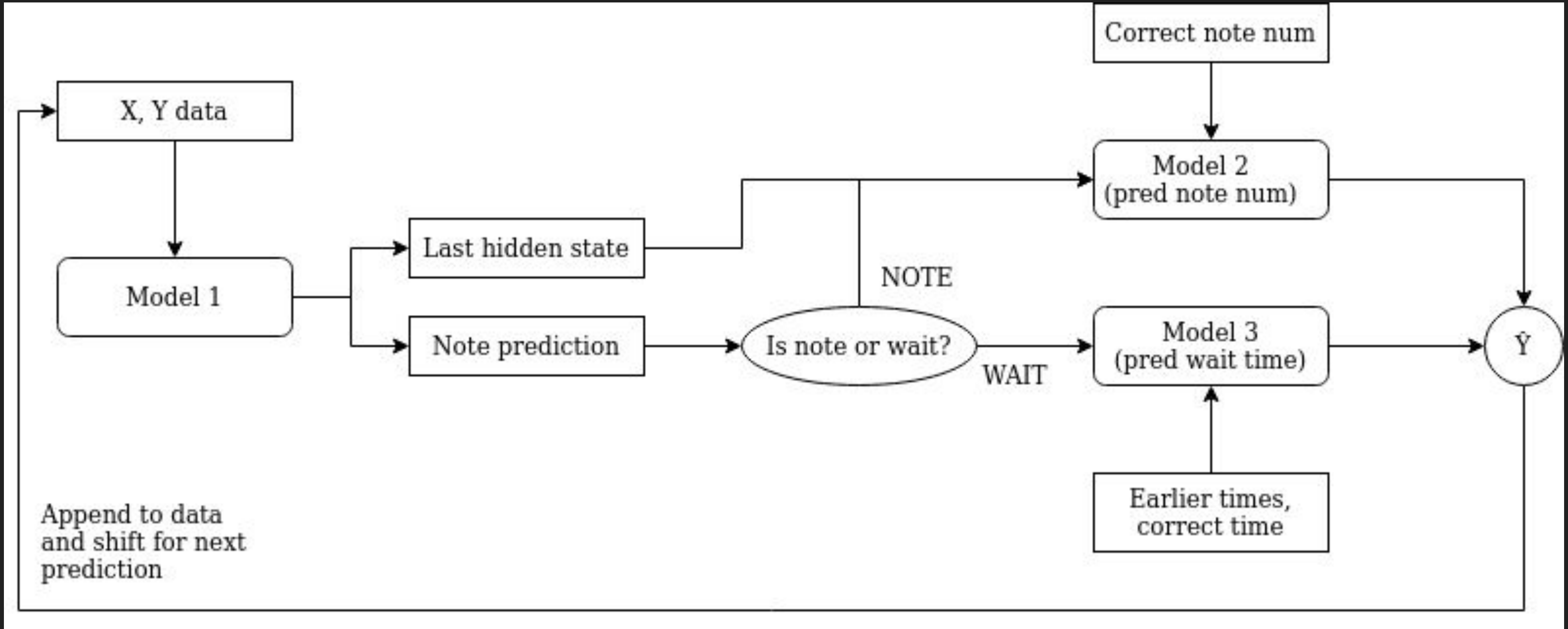
- FC Layer (162)
- Dropout(0.2)
- FC Layer

Model 3

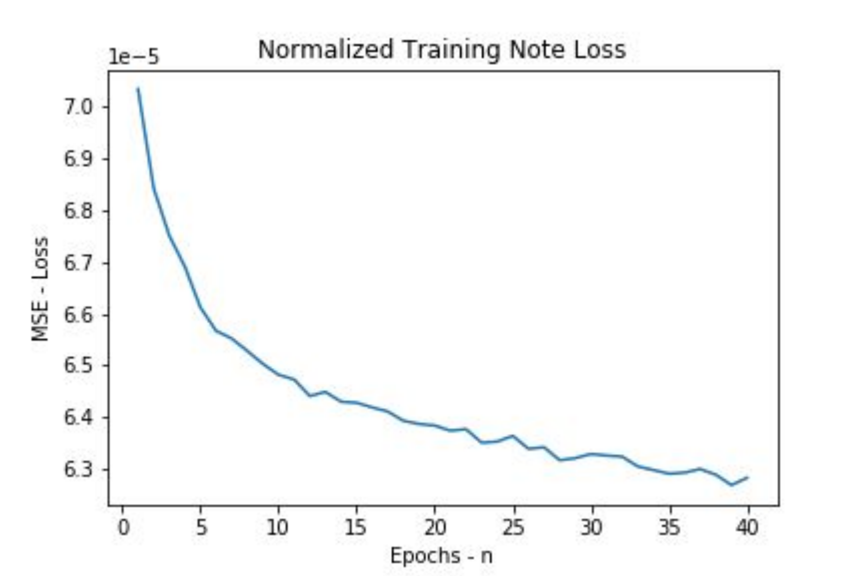
- FC Layer (128)
- Dropout(0.2)
- FC Layer

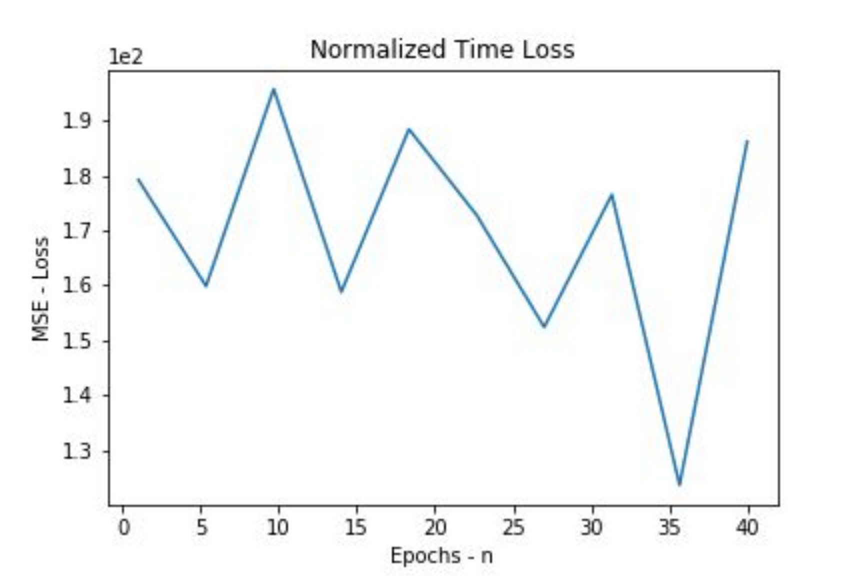
ADAM on models. MSE for model 3 (wait time) and CCE for other two.

Model structure (Laknes)



Losses MAESTRO model

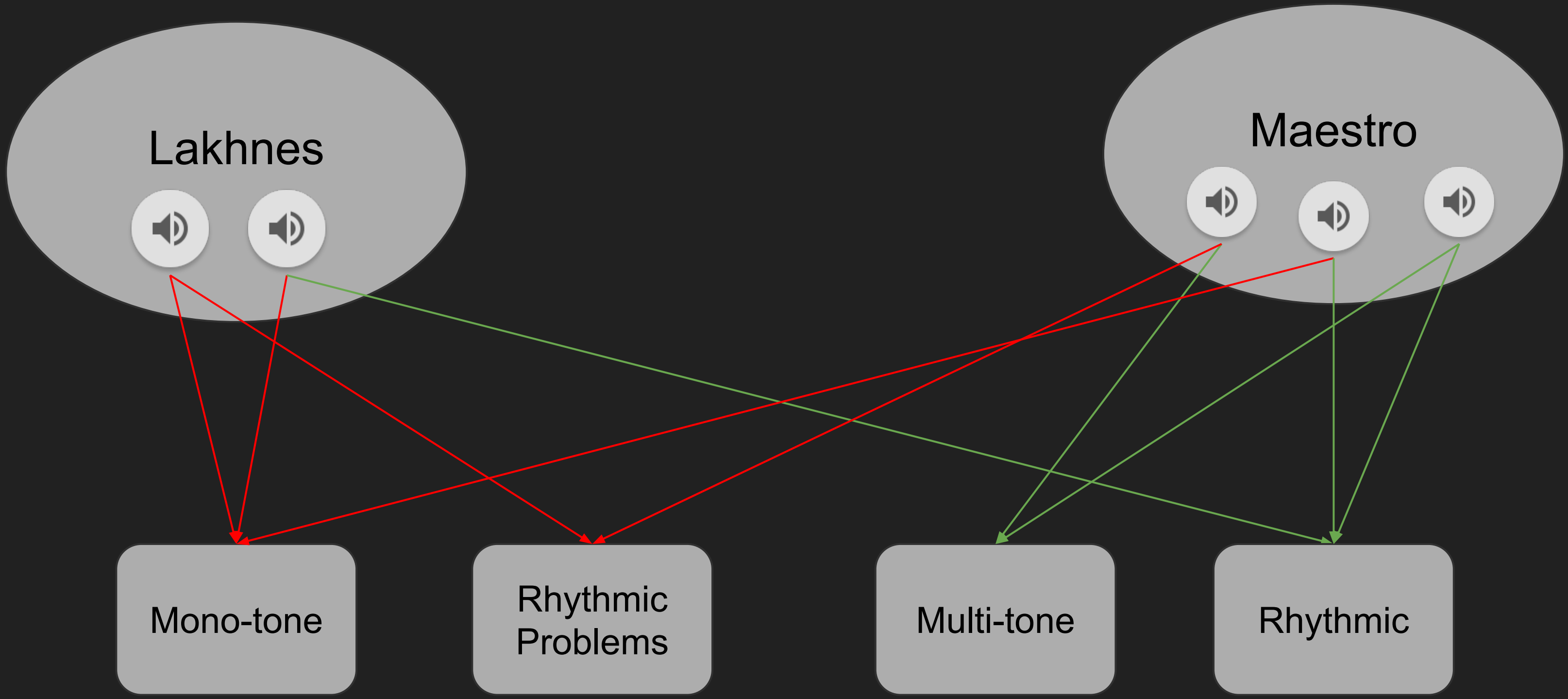




Losses Lakhnes

Dessa ser lite skumma ut, kommer typ inte säga så mycket om dem. Mer intressant att titta på.

Results


Lakhnes
Maestro
Mono-tone
Rhythmic Problems
Multi-tone
Rhythmic

Comparison of output, lakhnes

Predicted sequence (further)

P1_NOTEON_66
WT_523
TR_NOTEON_44
WT_710
TR_NOTEOFF
WT_639
TR_NOTEON_44
WT_581
TR_NOTEON_44
WT_591
TR_NOTEOFF
WT_567
P1_NOTEON_66
WT_530
P1_NOTEON_66
WT_629
TR_NOTEOFF
WT_633
P1_NOTEON_66
WT_592
P1_NOTEON_66
WT_761
NO_NOTEOFF
WT_721
P1_NOTEON_66

Predicted start (after green)

P2_NOTEON_90
WT_2208
P1_NOTEON_86
WT_35
P2_NOTEON_94
WT_24
TR_NOTEON_46
WT_2136
P2_NOTEON_86
WT_2208
P1_NOTEON_82
WT_31
P2_NOTEON_90
WT_2164
P2_NOTEON_82
WT_2212
P1_NOTEON_94
WT_31
P2_NOTEON_86
WT_2159
P1_NOTEON_66
WT_438
P1_NOTEON_66
WT_510
TR_NOTEOFF
WT_661
TR_NOTEOFF
WT_588
TR_NOTEON_44
WT_720
TR_NOTEON_44
WT_603

Analysis

Models tried: Many RNN/LSTM variants, VAE, seq2seq, seq2one

Not good! But not a complete failure

Time network: Common subdivision

Note network: Tends to stick to primed key

Things to try: Custom loss function, multidimensional LSTM, fix the VAE

Learning outcomes / conclusion

Ability to implement RNN/LSTM on other problems

Finding and implementing datasets is possible

Improved python skills

Generating good music using RNN/LSTM is harder then expected

Getting an output of some sort is not as hard

Generating music will give you a laugh.

Thank you for listening!