

Eberhard Karls Universität Tübingen  
Mathematisch-Naturwissenschaftliche Fakultät  
Wilhelm-Schickard-Institut für Informatik

## Bachelorarbeit Informatik

### **Neuronal nets and Music as an Art, with particular regard to GateL0rd [1].**

Fredrik Zeller

16. Juni 2023

#### **Gutachter**

Prof. Dr. Martin Butz  
Kognitive Modellierung  
Wilhelm-Schickard-Institut für Informatik  
Universität Tübingen

**Zeller, Fredrik:**

*Neuronal nets and Music as an Art, with particular regard to GateL0rd [1].*

Bachelorarbeit Informatik

Eberhard Karls Universität Tübingen

Bearbeitungszeitraum: 28.03.2023 – 01.06.2023

# Abstract

This thesis comes up with an experiment, in which several neuronal networks are tested according to predict a simple piece of music. Commonly neuronal nets are seen as black boxes, as it's impossible to comprehend the calculations which are causing a certain output.

To counter this problem, GateL0rds capacity to peek into this black box will be considered, as by means of it's gating structure GateL0rd (GL) allows to keep track partially of the hidden 'thoughts' of the net. It will be examined then if a correspondence between the musical content and GL's gating property shows up. Furthermore, to make the results comparable, a method of musical evaluation will be presented.

At the same time, this is a more generalised discussion about AI's <sup>1</sup> capacities to produce pieces of music, which can be attributed to the domain of art.

---

<sup>1</sup>Always supposed to be based on neuronal nets

# Contents

<b>1</b>	<b>Motivation</b>	<b>6</b>
1.1	AI vs. Author. Time and Space . . . . .	7
1.2	Music as Artwork. Notation and Construction . . . . .	9
1.3	Algorithmic Theory of Music . . . . .	10
1.4	Expression . . . . .	10
1.5	Now let's put the AI in Place of the Author! . . . . .	12
<b>2</b>	<b>The Aim of this Thesis</b>	<b>13</b>
<b>3</b>	<b>RNN, LSTM, GateL0rd</b>	<b>14</b>
3.1	Recap: RNN . . . . .	14
3.1.1	The cell . . . . .	14
3.1.2	Enrolling in time . . . . .	15
3.1.3	Examples . . . . .	16
3.2	GateL0rd (GL) . . . . .	17
<b>4</b>	<b>The Toy Experiment</b>	<b>19</b>
4.1	The Data . . . . .	19
4.1.1	Time-slices, chords . . . . .	20
4.1.2	The Data as input and output of the models . . . . .	22
4.1.3	Musical construction of the data . . . . .	23
4.2	The Models . . . . .	26
<b>5</b>	<b>Training</b>	<b>29</b>
<b>6</b>	<b>Musical evaluation</b>	<b>33</b>
6.0.1	Evaluation Data . . . . .	35
6.0.2	Results of musical evaluation over all checkpoints . . . . .	36
6.0.3	Single examples of prediction . . . . .	37
6.0.4	Summary, table of results . . . . .	38
<b>7</b>	<b>Gate L0rd's Lambda Parameter</b>	<b>40</b>
7.1	GateL0rd vs. harmonic progression . . . . .	40
7.1.1	GateL0rd model trained on extended dataset . . . . .	42
7.1.2	Modulation and B-major . . . . .	42
7.1.3	GateL0rd vs. Arpeggio (Monophonic) passages . . . . .	44
<b>8</b>	<b>Further work</b>	<b>45</b>
8.0.1	Cross Entropy . . . . .	45

8.0.2	Music Evaluation . . . . .	45
8.0.3	Peeking into Hidden States . . . . .	45
8.0.4	More complex Models . . . . .	46
<b>9</b>	<b>Conclusion on Musical Expression and Creativity</b>	<b>48</b>
<b>A</b>	<b>Toy song in B-flat major</b>	<b>53</b>
<b>B</b>	<b>Result Db-Major modulated to Eb-minor</b>	<b>55</b>
<b>C</b>	<b>Analysis of Bb-Major</b>	<b>57</b>
	<b>Bibliography</b>	<b>61</b>

## Chapter 1

# Motivation

“Is AI capable to create music?”

As we delve into the topic, we notice that this question is already broadly discussed, and multiple efforts have been done so far. [8] [12] [13] [14]

As it became evident, neuronal networks are very strong in recognizing and reproducing patterns. So we can restrict the area of discussion, and exclude music, which mostly relies on such patterns, like so called “elevator music”. This includes mainly music with functionality, for example, music for relaxation, more milk from cows, or music for warehouses in order to increase the sales volume. This type of music is already impressively mastered by AI. [18]

We note that the overall characteristic of this type of music is the absence of an author.

As a secondary theatre there exists a discussion about the question: Who is the author of AI-music? Is it the programmer of the neuronal net, or is it the owner of the train data ?

But our motivation to ask for the author is different. Instinctively we are looking for the author, because we are looking for music, which has got a ‘face’. AI-music - in the nature of patterns - is smooth.<sup>1</sup> While we are looking for wrinkles, fissures in that face, which leads later to the widely used terms of *expression* and *emotion*. Herewith it seems that we are located in the domain of so called arts-music.

“What’s art?” is a big and long lasting discussion, which can’t be done seriously at this place. Given only one hint to the famous essay „Das Kunstwerk im Zeitalter seiner technischen Reproduzierbarkeit “(Walter Benjamin 1935).

The underlying issue of this essay is the loss of the “now and here” called the “aura” of an artwork.[7]<sup>2</sup>

In this context the ‘composing AI’ can be considered as a severe amplification of the essay’s thesis, which deals more with photography and movie (cinema) as well with the technique to record music on vinyl, as at its time this were the most advanced

---

<sup>1</sup>One could even parallelize it with the mathematical idea of being differentiable in every point.

<sup>2</sup>„Noch bei der höchstvollendeten Reproduktion fällt eines aus: das Hier und Jetzt des Kunstwerks – sein einmaliges Dasein an dem Orte, an dem es sich befindet. ... Man kann, was hier ausfällt, im Begriff der Aura zusammenfassen und sagen: was im Zeitalter der technischen Reproduzierbarkeit des Kunstwerks verkümmert, das ist seine Aura. “

techniques of reproduction.

We could see in the actual context an intensification of the problem, which comes along with “the apotheosis of the machine”. The excessive idealization of AI systems, setting them on a good like level, touches nowadays almost all sections of human society likewise in a problematic manner. As we consider art as a sector of society, - and, in there, music as a another sub-sector, it seems to be an urgent point of interest, to reflect about music creation in the context of AI.

## 1.1 AI vs. Author. Time and Space

To work out the difference between AI-generated works and works by author (called *artist*), we oppose, as an example of contrast, the two work-types in the domain of visual arts.

One one hand there are AI generated images. Whole lists of ‘AI art -generators’ are available.[19] As we consider the results, first of all, from the point of view of art criticism, they commonly don’t pass a surrealistic “Post-Dalí-style”, which was taken anyway long time ago by the advertising industry, and such transferred from “where the art lives” to the domain of “every day life”.

Second the results are designed to survive in pure virtuality. They don’t *have to* leave the computer environment.

Of course it’s for example possible to print them out and organize a ‘real’ exhibition, but this is not the *character* of their creation, and would rather mean an overlaid step, in the sense of creating an art event, using AI -made photos. This would tend to mean ‘creating art’ - using the computer as any form of material.

So basically we miss something like the real physical impact on the recipient, the interaction of the work with his environment. This is then very similar to the term ‘loss of aura’ by W. Benjamin, mentioned above.<sup>1</sup> Probably the most important factor in ‘interaction with environment’ is the dimension of time.<sup>3</sup>

In reference to the AI-pictures, they obviously lack a clear time dimension. An AI-work could have been created at every time - given as start-time the existence of AI itself -, it can be seen at every time, or never, it can last for ever - given as end time the end of data conservation -, or it can be destroyed in its very same moment of creation. Its corpus has no defined extension in time.

Continuing the exhibition example: In contrast - working with computer materials... - Exhibitions have a clearly defined lifetime.

To get in touch with the recipient, there has to be an intersection of the time of the recipient with the time of the work. In case of AI-work the intersection is completely arbitrarily, not obligatory, what weakens crucially its interaction with environment.

---

<sup>3</sup>Here we force another bridge to music, where time is *the* significant dimension.

At least as important as time is the dimension of space, regarding the interaction of work with its environment.

Asking, what is the real corpus of an AI-work, and if we take it literally, the extension in space of an AI-work would be exactly the surface covered by the transistors that are actually needed to store its information. First this is minimal surface, compared to the surface of the viewer. Second this space remains somehow hidden. The work, does not show up, does not 'enter the room'. There is a second representation in space for the AI-work, which is its projection. Imaged on screen, canvas, whatever... , this space representation appears, like its above discussed time representation, arbitrarily: The projection could be 'very big', or could be 'none'. So the AI-work is facing in space the same problems it faces in time. Exactly like in the considerations concerning the time, there must be an intersection between the space of the observer to the space of the work. But as the extension in space of a AI-work is negligible small or arbitrary, so there is not necessarily such intersection, what weakens another time the AI-works capacity of interaction.

On the other side, we put works made by authors. As it is in the nature of things, in contrast to the AI-work, we are able to drop some example-names here. Let's take: 'Rembrandt', 'Marcel Duchamp', 'Christo', 'Banksy'. For every artist-work, we could now examine its behaviour in space and time in the same manner we did it for the AI-works above. For works of Rembrandt and mostly of Duchamp, its obvious the work in context of exhibition. Their extension in time and space is defined by the exhibition. So their momentum of interaction-with-environment is given by the condition of exhibition (museum).

The crucial point is: The interaction is obligatory. Once the recipient has decided to pass by, the interaction *will* happen.<sup>4</sup> Works by 'Christo' show clear extension in space. The interaction with their environment is even part of the idea. Their intersection with the space-time of the recipient is clearly given. Interesting point is the lifetime of such an object. Its predefined, not infinite and barely not repeatable. Works by 'Bansky' differ in behaviour in time. Its equally not infinite, but, when we think how such a work finishes his life, we are thinking more in terms of a sort of decay over time. For the space this would be equivalent to a fade-out behaviour. As we see: Time and space of an author-work are clearly defined, what results obligatory in an intersection, with the equally well-defined time-space-experience of the recipient.

---

<sup>4</sup>What are the exact sensations of the Viewer is not important at this place.



## 1.2 Music as Artwork. Notation and Construction

Music as an art, is music by authors. So let's draw a track by dropping some exposed names in a time line: We start with 'Perotinus Magnus' at Notre Dame (about 1200). Then we pass by the era of so called Baroque with 'Vivaldi', 'Bach' over classic - romantic composers 'Mozart', 'Beethoven', 'Wagner'. Finally we end up somewhere in our time with the order 'Schönberg', 'Stravinsky', 'Stockhausen', 'Boulez'.<sup>5</sup> This is the tradition of classical (european) music. This is the kind of music production which is worth to be considered in the context of the primary, higher ordered problem. The question of the beginning "Is AI capable to create music?" turns into the question "Is there any composing AI?"

To understand what could be a 'composing AI', we have to have a look at what the composers did.<sup>6</sup>

The recurrent theme is that all these composers were musicians, which, so to say, made music *indirectly*. Music became a set of formed, ordered, combined, modified and distinct elements. It turns out that music owns a *material*.

Every composition represents a distinct, unique state of this material and this state is fixed by what we call *notation*. Due the notation, the process of creating music passes a certain objectivisation. At the same time it becomes repeatable, all in the sense of an experiment in physics, and here the gap from art to science closes. Expressivity, personality - all the things, which are commonly expected to be part of a composition - should now be visible in notation, that is: The specific way in which the materials are used, where 'used' means something more complex than just a side-by-side arrangement of the elements. We think of any form of contextual interaction between the elements, which could be described for its part in a scientific-like way. We are entering here the domain of musical theory. In this regard numerous contributions were made during the centuries.[21][22] [23] [24] [25] Interestingly most of those, since Pythagoras, deal with the tonal system or more specifically with what is called 'the harmony'. As a loose parallel to the development of natural sciences, we observe a trend to capture more and more details in developing metrics to make things describable. About in the late 18th century there came in officially<sup>7</sup> the discussion of timbre or 'tone-colour'. [26]

Also in the sense of parallelism to the development of nature science, new theories claim to implicate previous theories. All previous work must be explainable by the new theory.[27]

Understanding the compositional process as a work based on *materials*, describing

---

<sup>5</sup>It's clear that the list is arbitrary and that it could be equally done by dropping other names.

<sup>6</sup>Or *do* - There are still existing some specimen of the species.

<sup>7</sup>'Unofficially' they did it always. See the more recent studies and performances of music from 1500 to 1700 by specialized baroque ensembles.

the whole music as a system of somehow referenced elements, is a quite contemporary approach, what, after all, not seems to be very surprising because again, it looks similar to newer scientific descriptions, which are bundled under the idea of 'systems theory'. Touched are fields like 'chaos theory' and 'modelling'.

## 1.3 Algorithmic Theory of Music

Because of this parallelism to natural science, it is furthermore not surprising, that in the recent decades, efforts were made, to model the field of musical theory, by means of computer programs. Various algorithms have been developed, to describe various aspects of music.

A whole theory (only for tonal music) on which these algorithmic approaches refer frequently, has been published by Fred Lerdahl and Ray Jackendorf in 1983.[28], which is again based on a not very modern 'page-turner' by Heinrich Schenker (1906).[29]

Computer papers of musical theory are starting consequently with some observation from that domain, in order to model their problem. 3 Examples:

- “A listener is thought to be able to organise musical notes into groups within musical streams/voices. A stream segment is a relatively short...”[30]<sup>8</sup>
- “Musical forms describe how pieces of music are structured. Such forms explain how the sections work together through repetition, contrast, and variation: repetition brings unity, and variation brings interest.”[31]
- “...analysis based on the notion of a chord complex. A chord complex is a labelled simplicial complex which represents a set of chords. The dimension of the elements of the complex and...” [32]

Overall, the entity of musical construction seems complex and almost unsolvable by means of traditional algorithms. Therefore the use of 'newer' calculation methods like neuronal networks, becomes totally understandable.

## 1.4 Expression

As well as popular as justified expectation towards music by authors is commonly called expression. It's expected, that the piece of music gets in touch with the recipient, causing some emotional effects. We defined in 1.1 something like the time-space-framework for it.

These effects are quite difficult to quantize or to measure. Apparently it's a matter

---

<sup>8</sup>No successful implementation was seen so far...

of neurobiology, where we would step in another wide scientific discussion.<sup>9</sup>

What remains up to us is to consider the 'cause' for such effects and we retire on 'how-is-the-music-made', because, if we break it down, on what is the object, what makes the composition to be expressive, we are otherwise ending up in the *materials*. In even commonly less complex music - rock and pop -, the series of Rick Beato, may serve as an example for the fact, that such consideration has to be done hard along the materials.[33] At long last it's the conviction that 'how-is-the-music-build' controls the emotional effects at the side of the listener.

Something similar is ment, when the (european) music discussion refers to the relationship between 'form' and 'content'. It's mentioned that 'content' does not exist without form. Even stronger: content is *made* by form.<sup>10</sup> A good example is this statement made by Anton Webern concerning (his teacher) Arnold Schönberg. „In Schönbergs Werken ist nur Musik,... Schönbergs Verhältnis zur Kunst wurzelt ausschliesslich im Ausdrucksbedürfnis. Seine Empfindung ist von versengender Glut; sie schafft völlig neue Ausdruckswerte, also braucht sie auch neue Ausdrucksmittel. *Inhalt und Form sind ja nicht zu trennen.*“[20]

That leads to the moment, where we can define 'expression' as a *quality of the state of materials*.

It's very similar to the thinking in the field of theoretical computer science, where the class of complexity 'NP' is defined by 'can be verified by a non deterministic Turing-machine'.<sup>11</sup> The size of the state space of musical material is infinite. (Do we know it ?)

A piece of music represents a single slice, a single specific snapshot from that state space. A certain order, relation, formation of what we define as musical elements, seems to cause the effect, we then call expression.

In particular we make responsible for this specific state the author, the composer of the piece. Obviously the author set up a special state of material, so that this state, performing the piece of music, triggers something in the brains of the listeners.

The question that follows immediately: Why specially *this* state of material ?

And this question can not really be answered.<sup>12</sup> Instead of an answer, there are only estimations on all possible things, influencing the author, starting with historical environment, passing by the capacities of the author, and ending with the authors health status.

Aggravating this situation, the effect itself has a historical dimension. We can't say

---

<sup>9</sup>As well we skip the entire discussion of how performance and interpretation do contribute on the expression-effects!

<sup>10</sup>According to experience, normally at this point comes up a misunderstanding. To be clear: We are not talking necessarily about music with text - e.g. a 'song'!

<sup>11</sup>In the corresponding lecture at Universität Tübingen (SS 2022), Professor Henning came up with the, since then, so called 'Kantatenproblem'.

<sup>12</sup>Many authors itself don't have an answer on this. They mostly don't know *how* they did it.

that a certain state of materials causes for sure at any time (and location!), the same emotional effects. We can't treat it as absolute.

Things, which were in the 18th century important, or kind of impressive, are now a days rather casual. Let's think for example of the 'Mannheimer Manieren', a bunch of orchestral effects, which were introduced at this time. A few years later the described effects were common language and no longer outstanding.

## 1.5 Now let's put the AI in Place of the Author!

If we want a composing AI, it consequently has to be able, same as the author, to generate that specific state of materials, which in every sense causes the effect of expression.

Now we demand the AI to be *creative*, creativity seen as the capacity to construct that state of materials. Usually with the word *creativity* comes along something like surprise, or unexpected. We subordinate these two terms to the pool of possibilities rather, than assuming that they are qualities of the author.<sup>13</sup>

---

<sup>13</sup>It remains a pure contextual question if musical events behave for example *unexpected*. It could be considered there as a principle of order.

## Chapter 2

# The Aim of this Thesis

We replace the author by AI. In practice, this means, that we are training a neuronal net, and after that expect, that the net is able to make predictions, which leads at the end to a piece of music (song).

In any case its about two-staged method:

- (a) Training.
- (b) Evaluation - the 'composing act' of the net. The new song is the result of an autoregressive process, in the sense that the output of the net is taken as the new input. So the result depends over time on itself.

In this paper we pursue multiple goals:

1. Does the approach work in principle? Do the predictions of the net (model) approximate what we expect?
2. What is the appropriate basics for the training process ? This comes along with some architectural questions about the loss-function and the output-function.
3. We want to have a look into the black box, by examining the learning process. We talk of *Partially Observable Domains*[1]. This is done, using an implementation of GateL0rd (GL), procured by Christian Gumbsch[5], which allows to have a look at the hidden states. We may gather information from there, *how / what* the model did learn.
4. How can we evaluate the results ? We want to be the evaluation inside the problem-domain. So we should find a way, to evaluate *musically*.

It seems important to point out that we work on music in symbolic notation, not on its acoustic appearance. That level of abstraction is needed for the composing process, as it was described in chapter 1.2.

As notation format, although lacking human readability, the standard midi format[11] is very suitable for computation. We will use midi files to create the input of the nets, and we will convert the output of the nets back to midi format, to obtain at the end 'real' audible songs.

## Chapter 3

# RNN, LSTM, GateL0rd

As this is about music, and music is completely related to a timeline, we have to use models, which are able to manage time series. We know that recurrent neuronal networks (RNN) could do this job.

## 3.1 Recap: RNN

### 3.1.1 The cell

A RNN is basically a cell, where the cell has two parts.

- (a) an **Input part**, where a new state  $H1$  is calculated.  
In case, the model contains more than just one cell,  $H1$  turns to *hidden / latent*.  
 $H1$  is a vector, written here as  $d \times 1$  matrix, where  $d$  is the size of the hidden state.

$$H1_{d \times 1} = U_{d \times n} \cdot X_{n \times 1} + (W_{d \times d} \cdot H0_{d \times 1} + B_{d \times 1})$$

Further:

$X$  := the input vector of size  $n$ .

$U$  and  $W$  := Matrices, where weights are stored.

$H0$  := the previous state (almost hidden).

$B$  := a Bias vector of the size of the input.

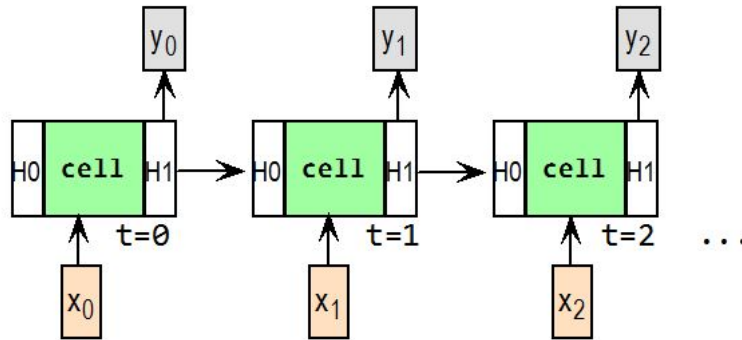
- (b) an **Output part**, which uses  $H1$  by multiplying it with matrix  $V$  and adding another bias  $C$ .

$$Y_{k \times 1} = V_{k \times d} \cdot H1_{d \times 1} + C_{k \times d}$$

Such a cell is also called a 'multi layer perceptron'.

### 3.1.2 Enrolling in time

Given the amount of time steps  $t$ , an input vector  $x_t$  corresponds to each time step. The cell is now 'multiplied'. The model will include as many cells as time steps required. The input part of every cell takes H1 from the previous call as H0 and the (by  $t$ ) assigned input vector as an  $X_t$  input.



**Figure 3.1:** time-enrolled RNN

This makes the network the 'deeper', the longer the input sequence  $X = [x_0, \dots, x_n]$  is.

Reccurency occurs at the moment, when a cell takes as part of its input information the output of the previous cell.

$\text{input}_t = h_{t-1} \circ x_t$  (where ' $\circ$ ' could be a concatenation)

This means that the net in every time step partially takes into account the results of the time step before. Finally in the last time step all the previous results will be contained in a vanishing manner: The further the time step goes back, the more diluted his information in the current time step. This uniform diminishing memories could turn into a problem. On one hand it's not evident, that this really corresponds to the structure of the input data. There might be time steps, not as important as others. On the other hand is given the possibility of an vanishing or exploding gradient on backpropagation, where the exploding gradient externally can be controlled by gradient clipping. But the vanishing gradient is immanent and could lead to the fact that calculating backward at certain point the weights of the following cells will not be updated, what means: The net probably does not learn properly.

To solve the problem of vanishing gradients, the Long Short Term Memory (LSTM) cell, was introduced as variant of RNN. In a LTSM cell we find a supplementary gate: The 'forget-gate' which determines how much of the corresponding time step should be considered in calculating the results. An cell state vector  $z$  is calculated. So that the value of a LSTM-cell could be approximated by:  $c_t \approx (c_{t-1} \otimes f_t) \oplus z$ .

Important is the additive behaviour with  $\oplus$ . It's the guarantee on backpropagation, that the gradient never goes to zero.

Furthermore the cell-state-vector could model some nuances in the importance of the time steps. One can imagine this like weighting the time steps.

### 3.1.3 Examples

An approach to generate music based on midi data via RNN was published on the page *tensorflow.org*.<sup>[12]</sup><sup>1</sup> The input of this net is a time ordered array of midi notes. One midi note is defined as a triple of  $(pitch, step, duration)$ , where  $step$  is something like the delta-time from the midi specification: The time interval from the beginning of a  $note_t$  to the beginning of the previous note  $note_{t-1}$ .<sup>2</sup>

All values in this note-triple are normalized.

The models configuration is:

1. Input layer.
2. The kernel, a LSTM layer of size 128.
3. Three separated output layers, corresponding to the triple of  $(pitch, step, duration)$ .

Every item of the note tripel has its own Loss-function:

*pitch*: A variant of Categorical Crossentropy<sup>3</sup>.

*step* and *duration*: A variant of Mean Squared Error (MSE).

Furthermore the model uses an 'ADAM' optimizer.

Another example is a paper by Eck and Schmidhuber. It discusses the advantages of LSTM over RNN, pointing out the same effect, we described above in the context of vanishing gradient: "In summary, LSTM is designed to obtain constant error flow through time and to protect this error flow from undesired perturbations." [9]<sup>4</sup>

We find the same discussion in *Music Generation Using LSTM and Its Comparison with Traditional Method*:

"LSTM has a property that the cell state gradient function is an additive function which is made up from four elements A(t), B(t), C(t) and D(t). ... This property

---

<sup>1</sup>The example fits well because, like our experiment, it's an approach on *notated* music.

<sup>2</sup>Delta time: In the Midi specification every midi event (Note off, Note on, Instrument change, Key change etc.) has a delta time, a time offset to the previous event, no matter what kind of event is the previous event.

<sup>3</sup>SparseCategoricalCrossentropy, produces the most likely index in a classification

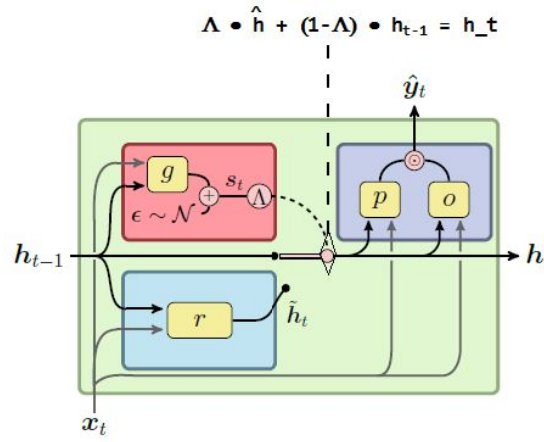
<sup>4</sup>We will refer to this paper later in the context of representation of data.



makes balancing of the gradient values better during backpropagation. LSTM updates and balances the four values, this additive property makes it very less likely for the gradient to vanish.” [10]

### 3.2 GateL0rd (GL)

GL consists of three subnetworks, which are called the *recommendation* net (r-net), the *gating* net (g-net) and the *output* net.



**Figure 3.2:** GateL0rd - Data flow [4]

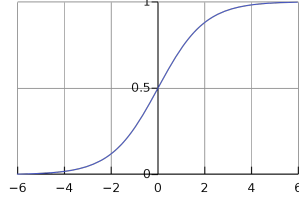
GL is handling the indifferent behaviour of latent states, proposing to every time-enrolled cell an alternative latent state by means of a gate. The basic idea is: If the new state is not really significantly changing - keep the old. [1] The actual latent state  $h_t$  is a mixture, between a new state, proposed by the recommendation-net, and the previous hidden state. A value Lambda ( $\Lambda$ ) is calculated, which determines how much parts of the proposed state  $\hat{h}_t$ , respectively of the previous latent state, make up the actual latent state. This is expressed in the following equation:

$$h_t = \Lambda_t \cdot \hat{h}_t + (1 - \Lambda_t) \cdot h_{t-1}$$

In this equation the extremes are:

- $\Lambda = 0 \Rightarrow$  new state is completely the old state
- $\Lambda = 1 \Rightarrow$  new state is completely the proposed state

Lambda itself is calculated as the result of a gating function  $\Lambda_t = G(i_t)$ , where  $i_t$  is the output of the gating network- a sequence on matrices with a Tanh-activation <sup>5</sup>. In the GL - paper [2] it's a point that this gate is *not* using the Sigmoid function  $f(x) = \frac{1}{1+e^{-x}}$ , because such a Sigmoid gate will never be completely dense. As we have  $\lim_{x \rightarrow \infty} = 1$  and  $\lim_{x \rightarrow -\infty} = 0$ , a minimum of information always drips over into the new proposed state  $\hat{h}_t$ . <sup>6</sup>



**Figure 3.3:** Logistic Sigmoid function.

---

<sup>5</sup>hyperbolic tangent function,

<sup>6</sup>We will work out later, that, for our purposes, in contrary the ReTanh-gate could be difficult, but on the contrary the Sigmoid gate has some better effect.

## Chapter 4

# The Toy Experiment

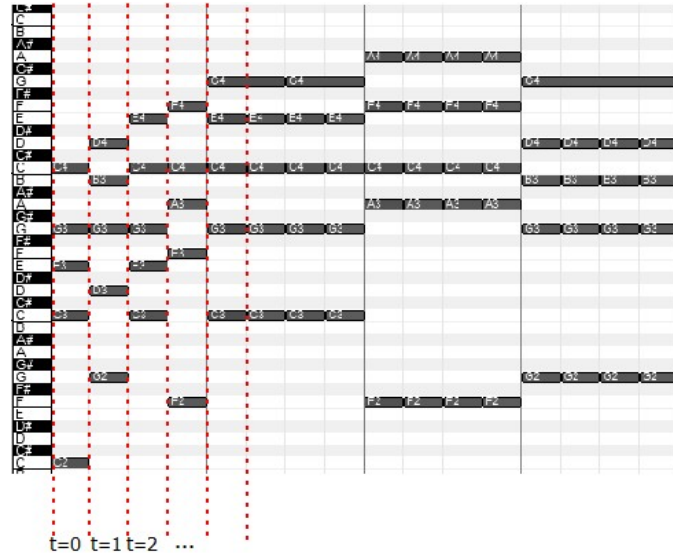
This is an experiment on a data set, which has very reduced musical content. The data set consists of a (big) number of short music pieces, generated by script in a music notation software. We call these, and also the net’s output, *Toy-songs*. (See example in appendix A.) The experiment is intended to show whether a process of learning and producing music, using the nets proposed below, is fundamentally possible.

### 4.1 The Data

A piece of music can be seen as a sequence of time-slices of any fixed length. We could call these slices then *chords*. Depending on the resolution, these chords could be equal to the ‘real’ chords of the piece, representing his harmonic structure. The chords may be randomly ‘contaminated’ which some additionally pitches, resulting for example from melodies, which used to occur exactly in its time slice. A similar method is mentioned by Eck, Schmidhuber : “The representation of time is straightforward, with one input vector representing a slice of real time. The stepsize of quantization can of course vary; if the quantization is set at the eighth note level (as it is for all experiments in this study) then eight network time steps are required to process a whole note.” [8] (pg.4). And it’s also proposed in *MUSIC TRANSFORMER: GENERATING MUSIC WITH LONG-TERM STRUCTURE* [15].

### 4.1.1 Time-slices, chords

To give an example, it seems intuitive to begin with a midi-rollout, representing one possible form of musical notation.



**Figure 4.1:** *Alle meine Entchen*, choral-like

All material of our Toy-experiment is made as a sequence of quater notes. This means, the resolution is exactly a quater-note.

We skip here the rhythmic problem mentioned above (4.1): Any note, with a duration longer than the resolution will be split up into  $n$  times the resolution - if resolution is a divider of duration. In this case the information about rhythmic polyphony is lost. As a subject to examine in future work, it should be found a method, that makes it possible to keep or retrieve this rhythmic dimension. <sup>1</sup>

<sup>1</sup> Loss of rhythmic information.

example:



resolved to quater durations:



Parts a) and d): No evidence for former half - notes.

Parts b) and c): It happened the contrary: Two eight-notes are summarized to the quater-slice. The order in time of the eighth - notes is forgotten.

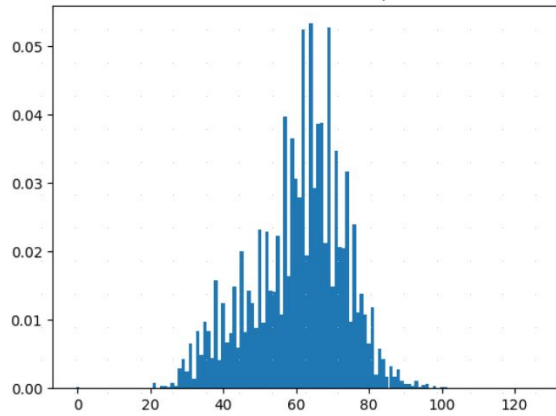
As shown in Figure 4.1, we build a sequence of binary vectors, where, every sounding pitch is a 1, the others remain 0. The midi specification allows pitches from 0 to 127, what leads to a vector of size 128 for every  $t$ .

A sequence of the length  $t$ , as we feed it as training-data to the nets, is then a 2-dimensional array with size  $(t, 128)$

The music in Figure 3 would the turn into:

$$\begin{aligned} &[(\dots, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 1, 0, \dots)]^T, \\ &(\dots, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, \dots)]^T, \\ &\vdots \\ &:] \end{aligned}$$

As this causes a lot of unused space, the length of the cord-vectors is reduced, by the *data limiter*.



**Figure 4.2:** Probability distribution of pitches, dataset

Based on a histogram of the complete data, during the loading process, the input vectors are trimmed:

Pitches at the borders (very high - very low), which anyway where never touched ( $P(\text{pitch}) = 0$ ), are skipped. This leads to a clearly smaller input size. Instead of the fixed 128 the value is floating around 60. The input size is then treated dynamically.

## 4.1.2 The Data as input and output of the models

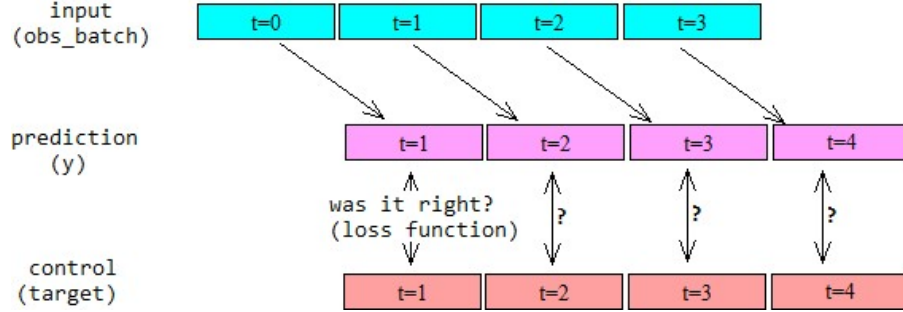


Figure 4.3: Training schema

The net is trained with sequences of  $n$  chords. Although we are ultimately only interested in the prediction of the  $n$ th chord, during the training process every chord of the sequence is presented and its prediction is judged with the loss-function.

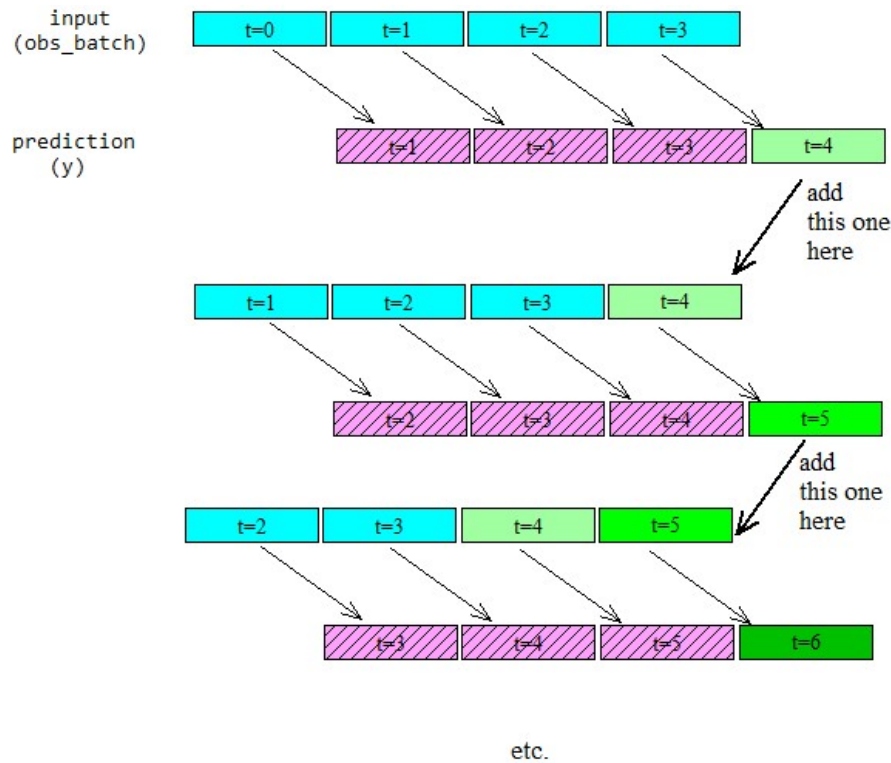
$$L_{all} = \frac{\sum_{i=0}^n \mathfrak{L}(c_i, y_i)}{n}$$

$c :=$  input chord

$y :=$  predicted chord

$\mathfrak{L} :=$  a loss function

For the 'composing' the net is send into an autoregressive process. Given a start-sequence of length  $n$ , this sequence will disappear, and after  $n$  steps the net makes further predictions on its own. Figure 4.4.



**Figure 4.4:** Prediction schema - net composing

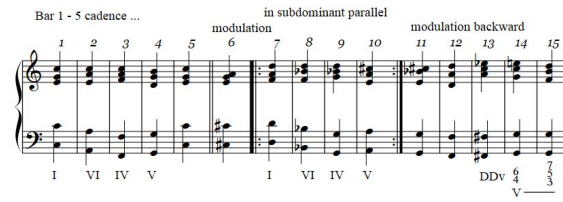
### 4.1.3 Musical construction of the data

Based on a limited chord material, any number of midi files is generated. Each file varies the *same* harmonic content:

- (A) Cadence in fundamental key.
- (B) Chromatic modulation to parallel of subdominant-key.
- (C) Cadence in parallel of subdominant.
- (D) backward modulation, using 'DDv'<sup>2</sup>

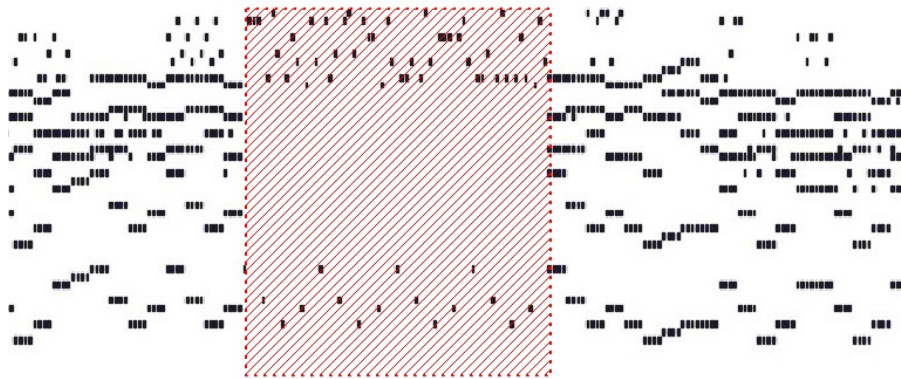
The order of the chords is fixed. The harmony changes happen every bar (measure 4/4). The elements A,B,C are repeated randomly in a certain amount. Section D occurs one time. Afterwards section A is repeated. Bases are doubled in octave, Some 'melody-notes' are randomly added, except in element D. This gives a type of 'melodic blur' to the data. The data is produced in all (12) keys. Later we will

<sup>2</sup>double dominant diminished



**Figure 4.5:** chord materials (in C major)

decide, how much of this data is necessary to train the net, in order to predict something in any other key. (See *APPENDIX A: 'Toy song in B-flat major'*)



**Figure 4.6:** Example: Midi roll-up with coloured 'Arpeggio period'.

On a second level a randomly selected part of the Toy song is reduced to 'one voice', shown in Figure 4.6. The resulting arpeggio gives a fundamental character change to the data during a certain period. The continuous homophonic structure is broken one time per song. The music switches from a structure of 'chords' - always 5 to 6 pitches on the same beat - , to 'melodic' - 1 pitch per beat -, and back.

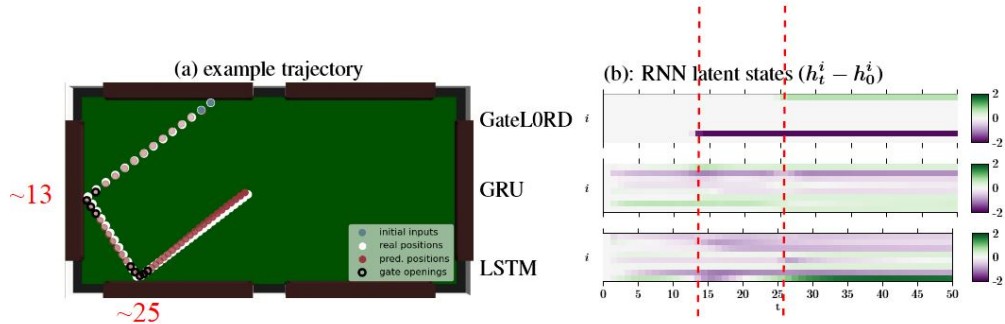
We are thinking here of GateL0rd, who's speciality is *the gate*. As described in the corresponding paper [1], the idea is, to trap fundamental changes in the conditions of the time sequence, and become less sensitive to little changes, which may be less important. See experiment „Billard“: Only when the ball is hitting the border



(chushion), there is a significant change in the latent state.<sup>3</sup>

Later we will see if we are able to somehow observe this behaviour in our Toy-experiment.

<sup>3</sup> Gumbsch, Butz: Supplementary Material [6]



## 4.2 The Models

### Loss function

Looking at our data situation, it turns out that we are ending up with a multiple-classification problem.

Consequently we choose 3 suitable Loss-functions.  
(For the following:  $x := \text{target}$   $y := \text{prediction}$  )

### Cross Entropy

The entries of  $x$  and  $y$  are considered as probabilities.

$$H(x, y) = - \sum_i^n x_i \cdot \log(y_i)$$

As we tweak into pytorchs implementation [17] , we notice, that on the  $y$  vector the Softmax function is applied, with the aim of summing up the probabilities to 1

$$\text{Softmax} : s(z_i) = \frac{e^{z_i}}{\sum_{j=i}^n e^{z_j}}$$

Pytorch-implementation of Cross Entropy:

$$H(x, y) = - \sum_i^n x_i \cdot \log(s(y_i))$$

In case the nets are using Cross Entropy, the activation function of the output layer is the *Sigmoid* function<sup>4</sup>.

### Categorical Cross Entropy

As the  $x$  and  $y$  vectors are simple binary, this seems the most suitable function. Pytorch-implementation (BCE loss) roughly:

$$l_i = y_i \cdot \log(x_i) + (1 - y_i) \cdot \log(1 - x_i)$$

---

<sup>4</sup>In the other 2 cases, we use the *Heayside Step* function

**Trivial Loss (0 - 1 Loss)**

Its a very simple idea to punish the net for every pitch, it did wrong.

If we subtract the input vector  $x$  (target) and the output vector  $y$ , take the absolute values of the entries in the resulting vector and add those up, we should obtain a meaningful number.

To obtain a function which is partially differentiable, this can be written as:

$$L_T = \sum_{i=0}^n (y_i - x_i)^2$$

with the summands  $l_i$

$$l_i = y_i^2 - 2x_i y_i + x_i^2$$

the partial derivative to  $y$

$$\frac{\partial l_i}{\partial y_i} = 2y_i - 2x_i$$

**Models overview**

To make some comparisons, the experiment is done on various models.

A group of 4 models will be used, as it was proposed by Christian Gumbsch.[5].

Every model consists basically of 3 components:

- a fan-in network
- the kernel
- a fan-out network. At its 'end' is located the output-function.<sup>5</sup>

To be comparable, the models are configured such that they have the same size, whereby the actual subject of interest is the kernel of the models.

1. In the first model the kernel is a LSTM net. From now on, we call this model **LSTM**
2. The kernel of the second model is a GRU net. It will be denoted as **GRU**
3. The kernel of the third model is a RNN net, denoted as **RNN**
4. In the 4th we set the GATELORD net as the kernel, named as **GL**

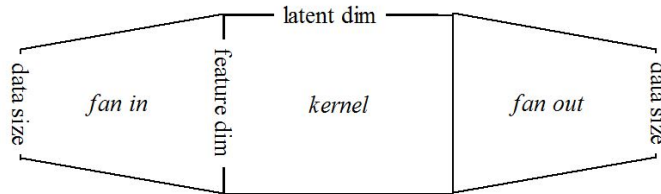
---

<sup>5</sup>*Heavy Side Step or Sigmoid*

After numerous and long lasting experiments, it turned out that it's sufficient, to choose a moderate size for all the models.<sup>6</sup>

As we conclude that the only sense of the fan-in-network is to shift the - dynamically calculated - data-size to the feature-size of the kernel, a single layer is sufficient for this purpose.

The same is valid for the fan-out-network: A single layer shifts the feature-size back to the data-size.



**Figure 4.7:** Schema: fan-in, fan-out

The main settings are:

latent dim	64
feature dim	128
layers internal (GL only)	1
preprocessing layers	1
postprocessing layers	1
learning rate	0,0005
optimizer	<i>Adam</i>

---

<sup>6</sup>Later on we will set up a model with doubled size.

## Chapter 5

# Training

As it is common practice, the data set was split in *training* data and *validation* data. For time reasons it was appropriate to train the bunch of models on a reduced data set. It was sufficient to figure out the main results, and to see on which model one should concentrate. The reduced dataset included  $\approx 54379$  sequences, where 65% were used for training, and 35% were used for validation. Each epoch run over the whole amount of training-sequences, resulting in a training-loss-value per epoch. Afterwards the model was tested on the validation-sequences, resulting in a validation-loss-value per epoch.<sup>5</sup>

The sequences were extracted from the Toy-songs in all keys, except B-major. We retained this key<sup>1</sup> with the aim to explore later, if the nets would be able to predict something in a unseen key.

All models were trained on a sequence length of 56 beats<sup>2</sup>. That means, referring to the data, the model had to predict 56 vectors. This range is required, to give the model some perspective into the 'future'. As the harmony changes happen mostly every 4th beat, this gives 16 changes to the model to learn something about the chord *progression*. In the data, the monophonic sections<sup>3</sup> are designed to last at least 16 beats, so that a sequence length of 56 beats gives a good chance to, - on one hand, span over, or - on the other hand, plunge into these sections.

Furthermore the training was interrupted with an *early-stopping*-mechanism, whose patience was set to 5. That means: After 5 epochs, the validation loss did not decrease any more, the training was stopped.

A variant of the model was saved after every training epoch, in order to be able to iterate over these checkpoints when dealing with the musical evaluation. (See 6)

## Training - results

Training results in the typical graphs, showing training-loss and validation - loss.<sup>4</sup>

---

<sup>1</sup> B-major is more or less random. It could have been another key out of 12 possible keys.

<sup>2</sup> see data: The music was given as a sequence of quarter notes

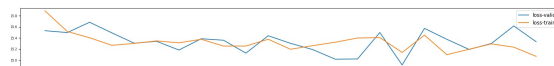
<sup>3</sup> only one voice

<sup>4</sup> somehow an inversion of what is called in other place *accuracy*.

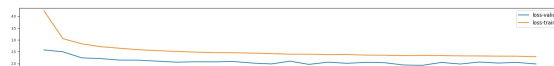
## Training / Validation-graphs under 3 different Loss-functions

- LSTM

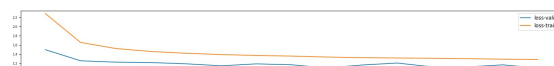
Cross Entropy:



BCE loss:

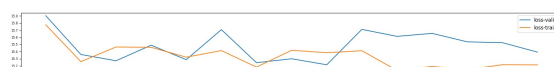


Trivial Loss:



- GRU

Cross Entropy:



BCE loss:



Trivial Loss:

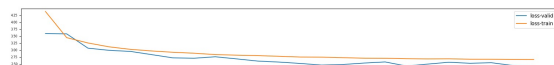


- RNN

Cross Entropy:



BCE loss:



Trivial Loss:

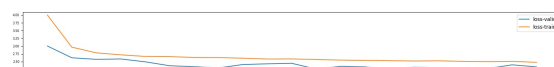


- GL (reg lambda = 0)

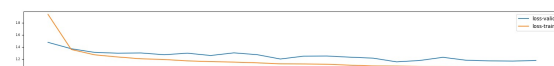
Cross Entropy:



BCE loss:



Trivial Loss:



For the cases Binary Crossentropy (BCE) and Trivial Loss we observe generally a diminishing behaviour, of the graphs for **evaluation loss**, and **training loss**, which means that all the models are learning to a certain point.

For BCE-loss and Trivial-loss the validation-loss is generally lower than the training-loss. (Exception LSTM with Trivial-loss and GL with Trivial-loss.) This is because in these nets are using for the training an extended version of the Heavy-side-step

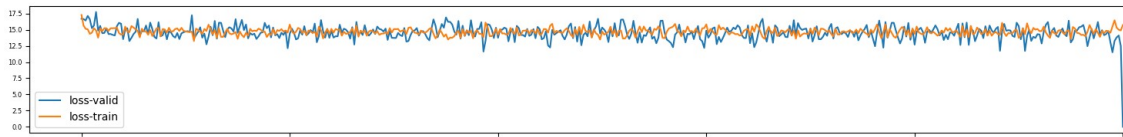
---

function, which uses a random calculation, meanwhile in the validation process they use the same function without the random component.

```
def forward(ctx, input):
    if (g_this_net.training == True):
        input_ = input - torch.rand(input.size());
    else:
        input_ = input - 0.5;
    return torch.ceil(input_).clamp(min=0, max=1);
```

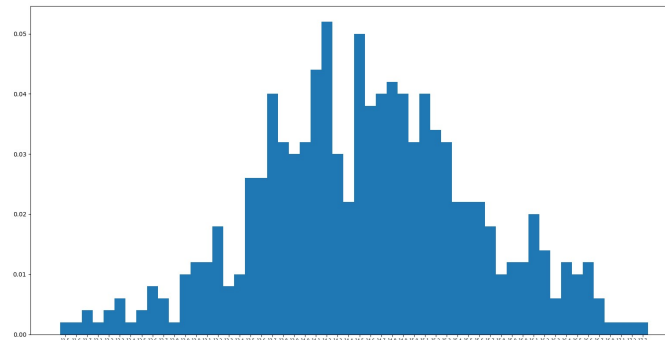
This leads to less accurate - because randomized - predictions in the training.

In contrast the graphs of the Cross Entropy are resulting 'zigzag'. An extended run over 500 epochs gave the following image: We get the impression, that the results



**Figure 5.1:** Graph over 500 epochs. Crossentropy

are random, what is confirmed, if we look at the distribution of values, which seems to approximate a normal distribution.



**Figure 5.2:** Distribution of Validation losses using Crossentropy (rounded values)

Later we will examine if the graphs, we obtain by musical evaluation, are tending to have the same diminishing course. Furthermore the minimal music-loss should be located more ore less at the same epoch as the minimal validation loss.

### Training GL with various values for parameter 'reglamda'

As one of the goals is to examine the behaviour of GL, it was trained with different values for the parameter reglamda, which regulates how much of the gating activity is added to the loss.

Calculating the loss. (Simplified version<sup>5</sup>):

1) A gated vector  $\Lambda$  is calculated.

$\Lambda = ReTanh(s)$  where  $s$  is the output of the g-net 3.2

2) The vector  $\Theta$  for the loss is calculated, using either the indicator function (realized with Heavy-side-step), or the Sigmoid function. In the GL-paper [3] the Heavy-side-step function is proposed, to calculate the  $L_0$  norm. Heavy-side-step is set to be the Indicator-function ( $\mathbb{I}$ ).  $\mathfrak{L} = \sum_{j=1} \mathbb{I}(\Delta h^j \neq 0)$

This results in a binary vector  $\Theta$ .

$\Theta = \mathbb{I}(\Lambda)$

3) Finally the mean of  $\Theta$  is taken, to obtain a number usable for the loss.

Here we observe a possible problem with Heavy-side-step gates:

In practice it could be observed that after few training steps all the entries in the vector  $s$  (output of g-net) are  $\leq 0$ .

This means that  $|\Theta|$  calculated by  $\mathbb{I}$  with the Step function is 0, which leads for all the consequent steps to a loss of 0, such there is no more effect on the training behaviour.

Therefore, instead of the Step-function, we favour in our experiment the Sigmoid function as the gate in 2).

The overall loss is then defined as  $\mathfrak{L}_{all} = \mathfrak{L}_{task} + \Lambda \cdot \mathfrak{L}_0$ .

For the value of  $\Lambda$  we chose a sequence, diminishing with the power of 10:  $\Lambda = 10^k$  with  $-3 \geq k \leq 0$  and  $\Lambda = 0$ .

---

<sup>5</sup> We skip here the stochastic quality of the gates.



## Chapter 6

# Musical evaluation

One of the the main question is: Do the predictions, the new songs, make sense in a musical manner ? How can we estimate this ?

Looking on every predicted song and estimating it's musical content (even as an expert), seems to be little trusting. The way to present the results to a group of test persons, how it was done here [16], could be successful on a large statistical scale, following the 'law of big number'. Both methods, at this point, are no option, so we have to look for another possibility.

At the end we want *one*-number which expresses the 'musicality' of the results, to make them comparable, so that we could for example determine which net performs best.

Do be able to measure the amount-of-musical-sense ('musicality') we have to invent some metrics to describe the musical facts. As we don't want to 'hang free', we strictly refer to the ground data. Simply because we *made* the ground data, we know what kind of musical sense they contain.

The metrics are:

1. Chord-hits % := Amount of valid chords. Relative to song length.  
Valid chords are mostly the chords, which are used in ground data. Such as:  
Major, Minor in basic-position;  
Major as six-four;  
Major<sup>-7</sup> (used as Dominant in ground data), 3 inversions allowed;  
Diminished 4 voices (used as Double Dominant in ground data).
2. Rate of Monophonic passages % := Amount of 4-grouped beats with less than 3 voices and valid bass note. Relative to song length.
3. Key coverage % := Amount of beats with well defined key. Relative to song length.  
In the ground data only basic-position was used outside of the modulation-areas. Therefore it's sufficient to observe the bass line, which should cover always the set of scale steps  $\{I, IV, V\}$ .
4. Modulation hits := Amount of modulations.

In the ground data always occurs exactly one modulation forward to the subdominant-parallel, and after that one modulation back to Tonic.

We give more freedom to the model in the sense, that it's allowed to invent more modulations, given that on forward modulation follows a backward modulation. We assume the maximal number of modulations as  $\frac{\text{song length}}{4}$  because one modulation lasts exactly 4 beats.

5. Rhythmic variance of bass := Number, describing deflection from regularity. The ground data is constructed in 4/4 bars. Hence the bass notes are changing every 4 beats, with exception of 'cadence-start-follows-on-cadence-end', in which case the bass change lasts 8 beats.

$d$  := a distribution of bass durations

$m$  := mode of the sample

$$rbv = \frac{\sum_{i=1}^n d_i \bmod m}{\sum_{i=1}^n d_i}$$

<sup>1</sup>

The corresponding value in the ground data should be 0.

6. Variance of chords := Number, describing how much different cords are used. Using a histogram of the sample, the *variance* is calculated. A reference value is calculated from the ground data.

Using these metrics, an over-all-loss per model is calculated as the averaged sum of the single losses (1 - 6). The lower the over-all-loss, the better the *musicality* of the song, where every loss should be a number between 0 and 1.

1. Chord-hits:  $0 \leq \text{loss}_{\text{ch}} = \frac{\text{lengthSong} - \text{chordHits}}{\text{lengthSong}} \leq 1$

2. Rate of Monophonic passages: Is calculated against the corresponding value from the ground data.

$$0 \leq \text{loss}_{\text{mp}} = |\text{mp}_{\text{ground}} - \text{mp}_{\text{song}}| \leq 1$$

3. Key coverage:  $0 \leq \text{loss}_{\text{key}} = 1 - \text{keyCoverage} \leq 1$   
If Key coverage is complete, the loss is 0.

---

<sup>1</sup>The mode, most frequent value of a sample, in this case simply 4.

---

4. Modulation hits:  $0 \leq \text{loss}_{\text{mod}} = 1 - \frac{\text{numMods}}{\text{maxMods}} \leq 1$

5. Rhythmic variance of bass:  
 $0 \leq \text{loss}_{\text{rvb}} = \text{rvb}_{\text{song}} \leq 1$

6. Variance of chords: Is calculated against the corresponding value from the ground data.  
 $0 \leq \text{loss}_{\text{vch}} = |\text{vch}_{\text{ground}} - \text{vch}_{\text{song}}| \leq 1$

The over-all-loss per song is then calculated with:

$$\text{loss}_{\text{all}} = \frac{1}{6} \cdot (\text{l}_{\text{ch}} + \text{l}_{\text{mp}} + \text{l}_{\text{key}} + \text{l}_{\text{mod}} + \text{l}_{\text{rvb}} + \text{l}_{\text{vch}})$$

## 6.0.1 Evaluation Data

The evaluation data consists of a bunch of Toy-songs.

Its divided in two categories: *Seen* and *Unseen*.

*Seen* contains approx. the half of the actual training data. *Unseen* contains approx. the same number of songs which were *not* presented for training, including a certain amount of songs in B-major, which was the key, by what the nets were not trained. We run every model on checkpoints, saved during the training process. Ideally the checkpoint, which was said to have the minimal validation loss, should have the minimal loss of musical evaluation.<sup>2</sup>

Given the beginning of a Toy-song - 16 beats as starting sequence<sup>3</sup> - the net's core task - *compose* - is to add 72 more beats to it. The whole is considered then as the 'new song', and presented to the musical evaluation.

```
forall checkpoints do
  newsongs
  forall seen and unseen do
    | newsongs += compose(startsequence)
  end
  musiceval(newsongs)
end
```

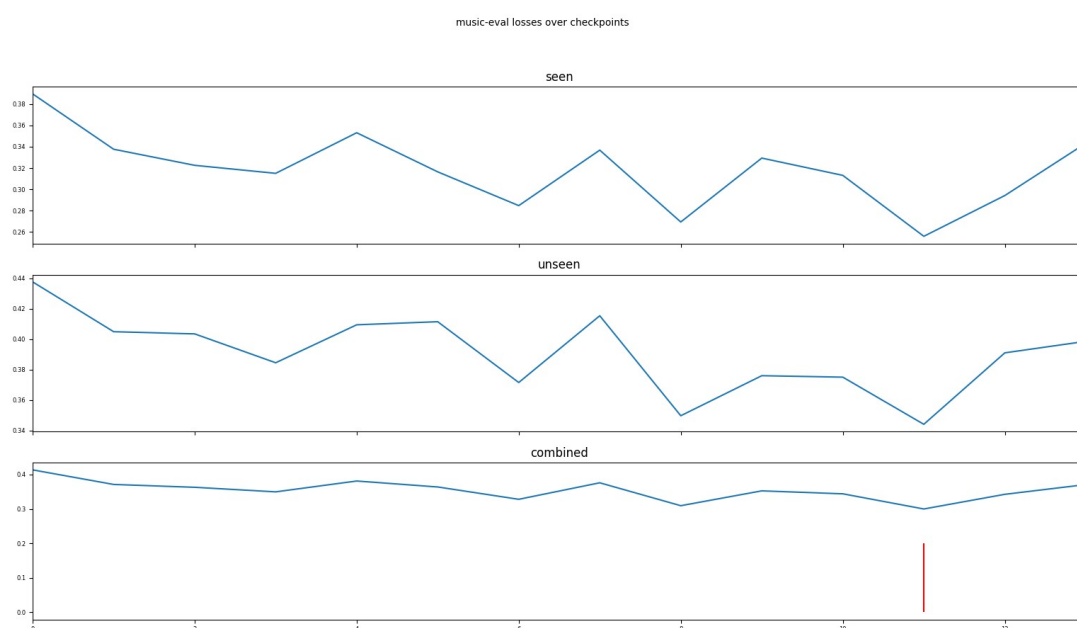
---

<sup>2</sup>unfortunately not always ...

<sup>3</sup>Longer makes no sense. Because then you don't need a net, you rather compose it on your own.

## 6.0.2 Results of musical evaluation over all checkpoints

Applied on a whole series of validations of a model, the resulting graph of musical-evaluation-loss and the graph for training validation, should behave somehow in the same way,- they should decrease. As we saw (5 ), the models which use BCE-loss or Trivial-loss are performing correctly. Therefore we choose as an example LSTM trained with Trivial-loss.



**Figure 6.1:** Example 1: Musical evaluation. Loss decreasing over checkpoints. LSTM using Trivial Loss

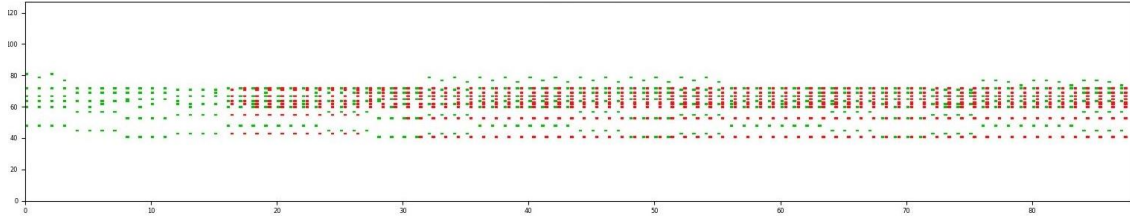
The minimum loss of musical evaluation is at epoch 11, which corresponds more or less to the value, indicated by the training process: 'Minimum validation loss at epoch 8'. See the expectation in 6.0.1.

Over all we observe the same decreasing behaviour as in the graphs of training/-validation - loss.

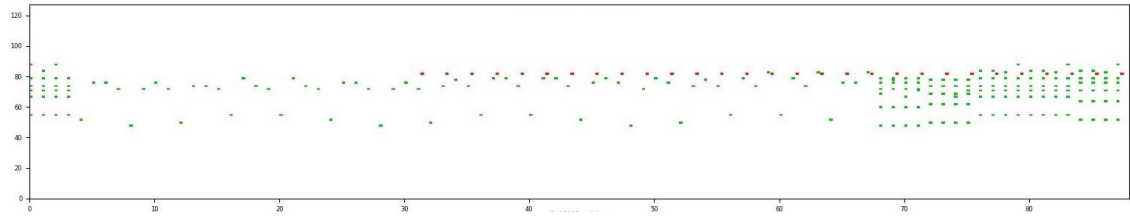
### 6.0.3 Single examples of prediction

Running one specific checkpoint on the evaluation data results in data specified for every song.

Two examples:



**Figure 6.2:** 'Good' prediction on file '0ModArp\_198'(C-ma), GL  $\Lambda = 0.001$ , Sig-moid gates, Crossentropy loss



**Figure 6.3:** 'Bad' prediction on file '7ModArp\_1' (G-ma), GL  $\Lambda = 0.001$ , ReTahn gates, Trivial loss

Figure 6.2 shows a 'good' prediction, Figure 6.3 a 'bad' one. In Fig. 6.3 no chords were predicted. The model immediately generated an 'ostinato' on Bb5, which although was interpreted as a monophonic passage. The musical losses are reflecting this.

loss type	Figure 6.3 (bad)	Figure 6.2 (good)
chord hits	0.954545454545455	0
monophonic passages	0.229839930493661	0.229839930493661
key coverage	0.920454545454545	0
modulation <sup>a</sup>	1	1
rhythmic bass variance <sup>b</sup>	0	0
chord variance	0.0197877002616428	0.0120090241018006
overall	0.52077127179255	0.20697482576591

<sup>a</sup>No modulation in both cases.

<sup>b</sup>There seems to remain a misconception in the calculation. See: 8.

## 6.0.4 Summary, table of results

LOSSFUNCTION:	binary crossentr.	crossentropy	trivial
LSTM			
min validation-loss	1.92507	14.91465	1.10372
at epoch	20	16	8
min music.eval-loss	0.26610	0.25864	0.30004
at checkpoint	11	15	11
RNN			
min validation-loss	2.42850	15.04300	1.29282
at epoch	17	16	17
min music.eval-loss	0.31341	0.27170	0.31623
at checkpoint	18	12	22
GRU			
min validation-loss	2.17030	15.21624	1.18914
at epoch	21	8	15
min music.eval-loss	0.32324	0.26185	0.33685
at checkpoint	10	10	4
G.LORD $\Lambda = 0$			
min validation-loss	2.147900	15.04836	1.15199
at epoch	19	4	15
min music.eval-loss	0.33560	0.24635	0.31781
at checkpoint	19	4	15
GL $\Lambda = 0.0001$ Sigmoid-gates			
min validation-loss	-	15.03444	-
at epoch	-	18	-
min music.eval-loss	-	0.22972	-
at checkpoint	-	14	-
GL $\Lambda = 0.001$ Sigmoid-gates			
min validation-loss	2.05554	15.16514	1.15462
at epoch	21	2	19
min music.eval-loss	0.31970	0.23614	0.32306
at checkpoint	19	5	18
GL $\Lambda = 0.01$ Sigmoid-gates			
min validation-loss	1.94724	14.88191	1.14446
at epoch	46	6	18
min music.eval-loss	0.32185	0.24499	0.31249
at checkpoint	22	11	21
GL $\Lambda = 0.1$ Sigmoid-gates			
min validation-loss	2.32009	15.07786	1.20030
at epoch	6	8	17
min music.eval-loss	0.32779	0.26144	0.34623
at checkpoint	4	13	13
GL $\Lambda = 1$ Sigmoid-gates			
min validation-loss	2.76168	15.59271	1.70832
at epoch	12	15	8
min music.eval-loss	0.31597	0.22842	0.34460
at checkpoint	10	5	7
GL $\Lambda = 0.0001$ ReTanh-gates			
min validation-loss	-	14.90424	-
at epoch	-	6	-
min music.eval-loss	-	0.24368	-
at checkpoint	-	9	-
GL $\Lambda = 0.001$ ReTanh-gates			
min validation-loss	2.34398	15.22307	1.27749
at epoch	2	0	3
min music.eval-loss	0.35642	0.25125	0.37870
at checkpoint	1	4	4

---

The table shows that the models, using crossentropy-loss are ending up with the lowest musiceval-loss. But as mentioned in chapter 5, (See 5.1) we suspect this to be a random result. On the model GL  $\Lambda = 0.001$  ReTanh - gates, happened that the minimal validation loss occurred at epoch 0, which is a strange thing, because the model could not 'know' anything at this time. We take this as another indicator for the randomness.

Further we observe that the only place, where the validation and the musical loss are located exactly at the same epoch, is in GL  $\Lambda = 0$ .

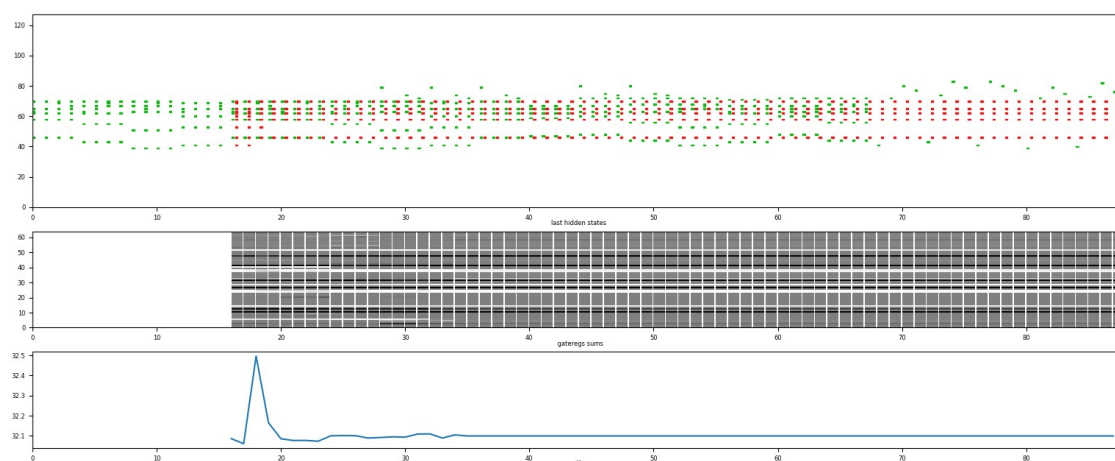
Considering all results so far, they tend to be 'rough'. On their best, they are just repeating the correct basic cadence. We would like to see more details like the supplementary melody notes, the modulations or the monophonic passages. Therefore a supplementary model in double size is trained on a extended dataset.

latent dim	128
feature dim	256
Lossfunction	BCE

## Chapter 7

# Gate L0rd's Lambda Parameter

The amount of  $\Lambda$  influences the predictions in the following way: As it is the task of  $\Lambda$  to ignore little changes, at the extreme, ( $\Lambda = 1$ ), all harmonic changes are considered as 'little', which results in convergence to one chord. We could say: The higher the  $\Lambda$ , the more rigid the prediction.



**Figure 7.1:** '-2ModArp\_5'(Bb-ma), GL  $\Lambda = 1$ , Sigmoid gates, Crossentropy loss.

After a period of wriggling in the gates, a constant value is established. This results in convergence to a Bb-major chord, which is luckily the tonic of the song.

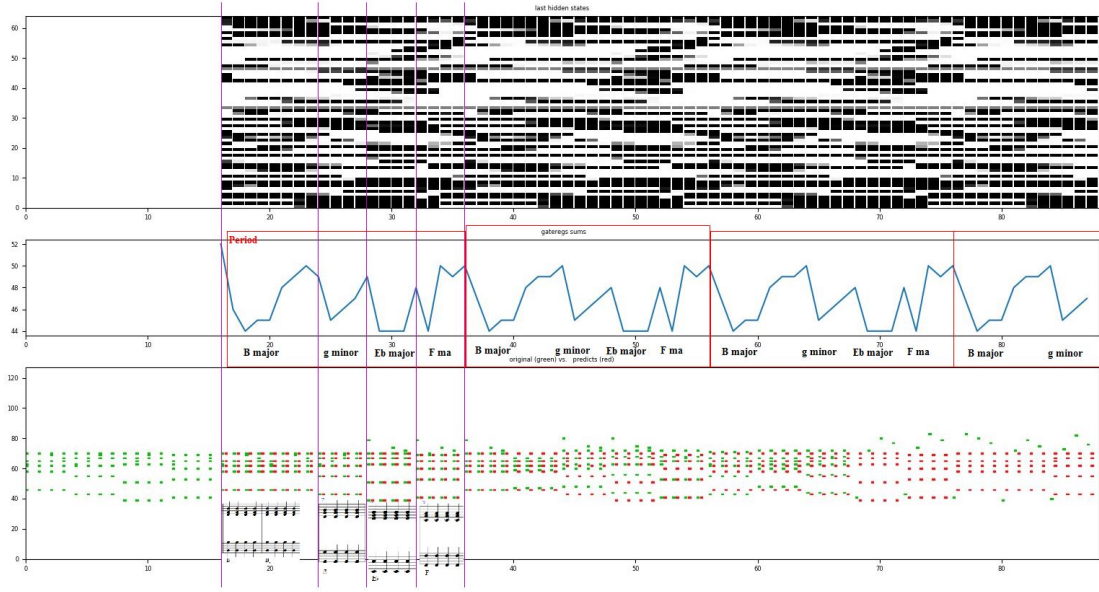
We observe that this convergence-effect seems to amplify, if we use ReTanh gates instead of sigmoid gates, as they are more dense and insofar more strict in behaviour.<sup>5</sup>

## 7.1 GateL0rd vs. harmonic progression

Model: Gate L0rd.  $\Lambda = 0.001$ . Gate-type = ReTanh. Trained on reduced dataset.

There obviously exists a period of gate-reg sums, which is clearly corresponding to

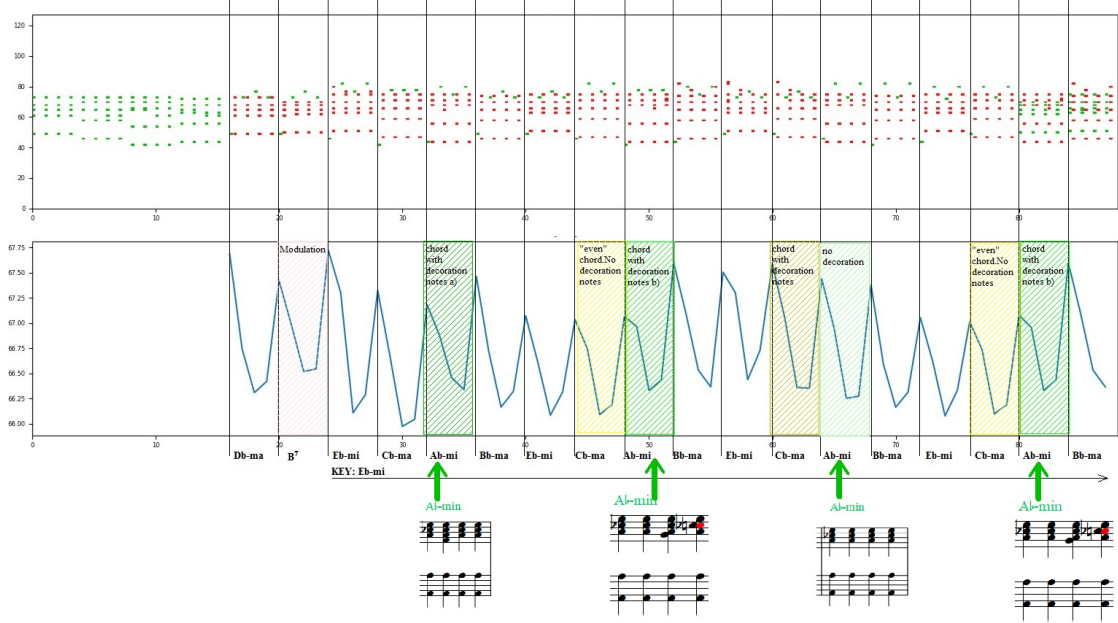




**Figure 7.2:** predicted song -2ModArp5

the harmonic period. The same period can be observed in the sequence of last-hidden-states. With a clear graphical correlation, we verify the regularizing effect of the gates on the last-hidden-states.

### 7.1.1 GateL0rd model trained on extended dataset



**Figure 7.3:** extended model: GL. with BCE-loss. Predicted song: 1ModArp1

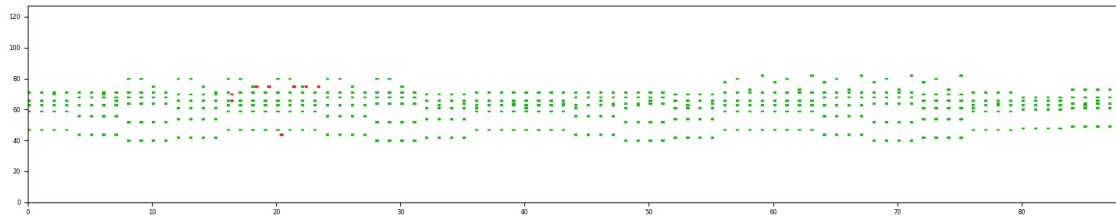
As expected the result shows now more details. See: **Figure 7.3**. Particularly we see some decoration notes, which we introduced in 4.1.3. If we split up the graph of gate-reg-sums barwise, we observe an superimposed pattern. Each bar-pattern mainly takes in count the actual harmony (Eb-minor, Cb-major etc.), and seems to be modified by the decoration notes. If the harmony is the same *and* the decoration is the same, the pattern is the same. This can be understood by observing for example the Ab-minor chord, where on the last beat the model obviously couldn't decide between major and minor, which results in a 'wrong' decoration-note. A complete harmonic analysis of this result can be found in appendix B.

### 7.1.2 Modulation and B-major

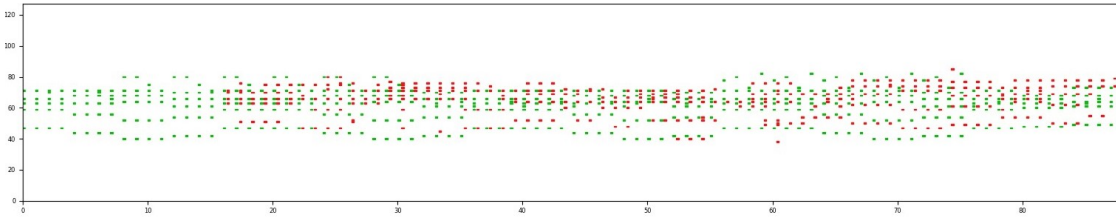
At the border of the experiment, we want to know if the model could predict something in a key, which was not presented for training. Therefore B-major was retained. Surprisingly the results are poor: No model was able to produce a valid output in B-Major.

Some failure examples: (Detailed analysis in appendix C)

extended GL model BCE  $\Lambda = 0.0001$ : Not really a prediction.

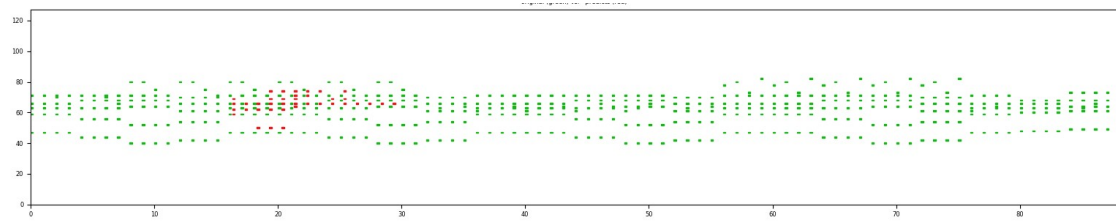


RNN BCE: Ended up in D-maj, wandering around B-minor, taking then B-minor as parallel to tonic D.



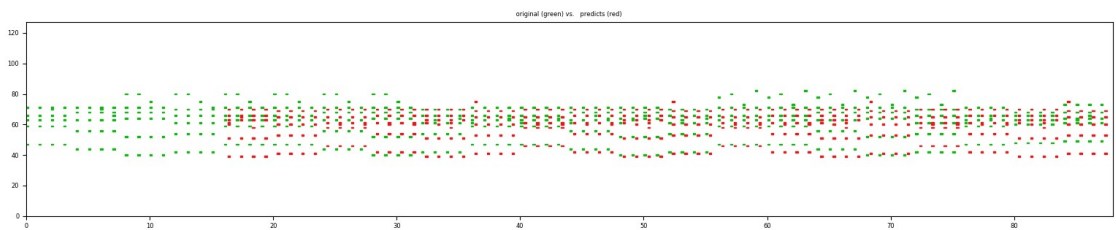
LSTM Trivial-loss: Poor prediction.

Passing over single chord Bb-minor, directly to D-major.



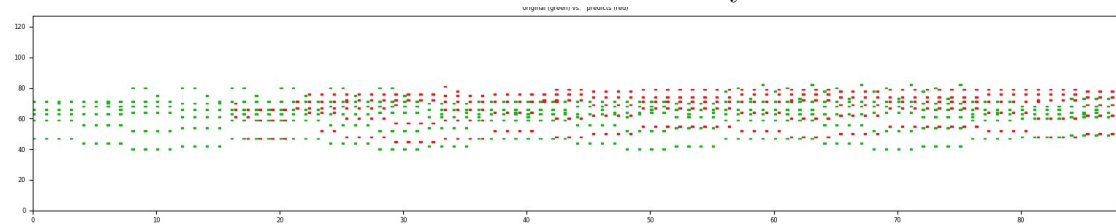
GL model BCE  $\Lambda = 0.01$ : We admit a fancy modulation to B-minor:

D $\sharp$ -mi (subdominant-parallel) is reinterpreted as subdominant in Bb-minor.



GL model Cross-entropy (!)  $\Lambda = 0.001$ : Uses directly the B-major chord to switch to E-minor

This is a classic diatonic modulation: Tonic of old key becomes dominant of new key.

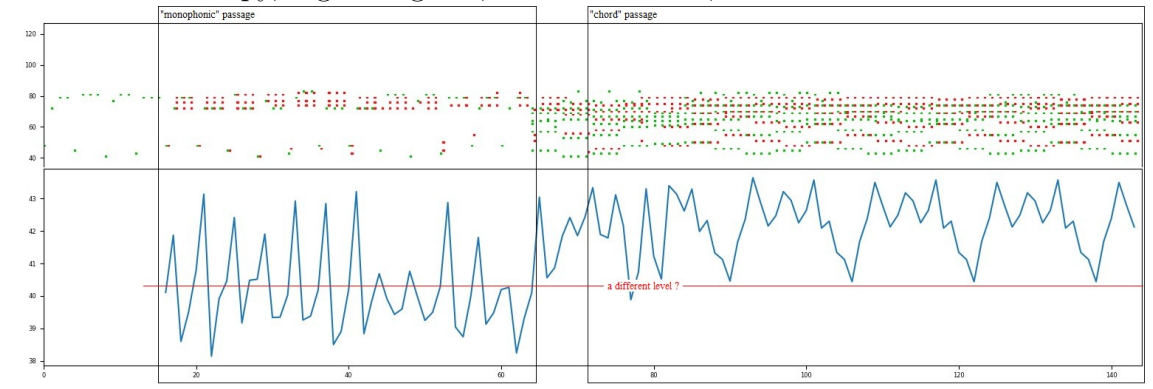


It was not possible to any model to do an abstraction over the key. Mostly the models assumed a D-major environment, where B-minor is the parallel to tonic, as they learned with the ground-data. In a period of disorientation they took B-major for B-minor, and did what they learned. The models learned harmonic progression and modulation, not transposition. We observe that they try to solve the transposition problem with modulation - e.g. reinterpret the unknown chord as a known element. We deduce there as a general behaviour, that after a period of confusion the models plunge straight back into what they learned. This is an important point in further considerations. 9

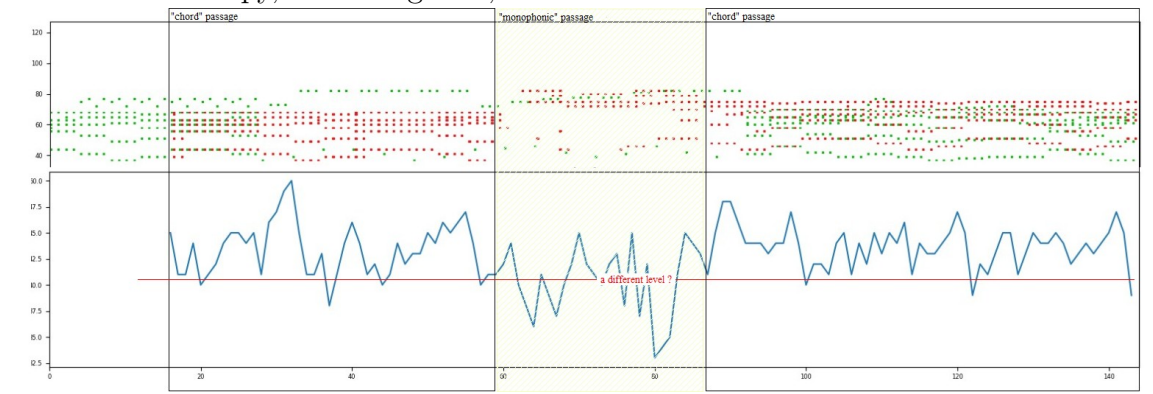
### 7.1.3 GateL0rd vs. Arpeggio (Monophonic) passages

Considering the results, there is no real evidence for a correspondence between the predictions on arpeggio-passages and the gating activity. We observe that the gate-reg graph is somehow changing, like operating on a different level, but it's not possible for now to determine something exact.

GL Cross Entropy, Sigmoid-gates, features = 256 ,  $\Lambda = 0.0001$ :



GL Cross Entropy, ReTanh-gates ,  $\Lambda = 0.001$ :



## Chapter 8

# Further work

### 8.0.1 Cross Entropy

It turned out that - once the cross-entropy-models hit randomly a minimum at the validation loss, from there they produce better results, than the rest. The Cross-entropy-models seems to be more adaptive, more elastic, which is understandable, due the fact that Cross-entropy works with probabilities instead of 'hard' binary. So we should try to save that, and solve the random problem which occurs during the training using Cross Entropy<sup>5</sup>. Maybe we should get rid of the Softmax function in the pytorch implementation 4.2 and implement simply  $H(x, y) = -\sum_i^n x_i \cdot \log(y_i)$

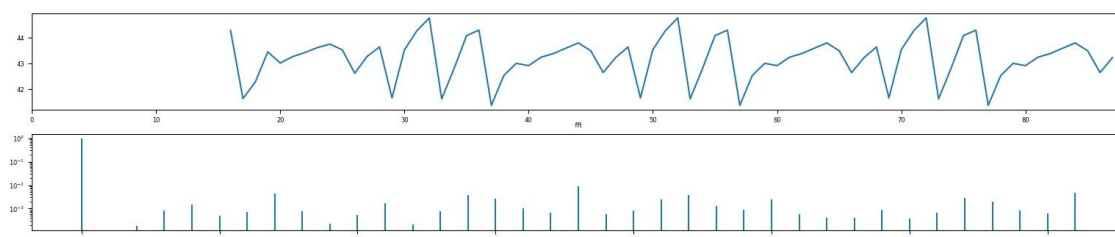
### 8.0.2 Music Evaluation

There remain some concrete things to do:

1. The problem of bass-variance can't deal with the case: 'There is no bass'. In this case it should not result a loss of 0. *b*
2. Convergence in the harmonic progression should be punished. Concerning this matter another criteria should be developed.

### 8.0.3 Peeking into Hidden States

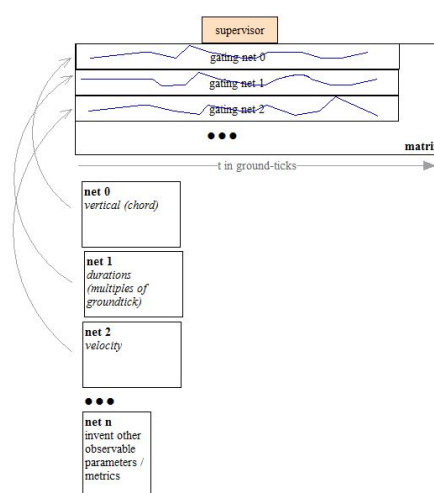
It would be interesting to do a deeper analysis of the '*thoughts*' of GateL0rd. Is there a way to interpret the graph gatereg-sums, beside the visual recognition of patterns ? Maybe a Fast Fourier Analysis could help to discover such patterns. See: **Figure 8.1**.



**Figure 8.1:** FFT transformation of GateL0rds 'brainstream'

## 8.0.4 More complex Models

- 1.) A first step would be to construct a GL-model with several layers. One could examine the gate-regulation for each layer. We assume that every layer is responsible for a specific task. E.g.: One layer is managing the harmony while another layer may deal with the number of voices (monophonic passages), and so on. Musical 'sense' would be readable more clearly.
- 2.) If the gates are analyzable, they are manipulable, what imposes the question, if the manipulation of the gates could make sense in an extended, modular model. Imaginable a net in form of a tree with a kind of gate-supervisor. We would have various subnets, which are dedicated to different tasks. The supervisor would be trained by the gatings of these subnets. In reversed process - aka 'composing' - the supervisor would determine the gatings for each subnet.



**Figure 8.2:** mock up: supervisor-net

---

3.) In the end we are thinking of building a neuronal net, which is able to produce *style copies* of a certain type of music, where the interesting point would be to copy the personal style of a concrete author, like it's commonly done manually as an exercise at the conservatories of music. It would be something more modest than creating art, because we remain concious, that we are creating *imitations* of some already invented music.

## Chapter 9

# Conclusion on Musical Expression and Creativity

We defined Expression as a quality of state of the material. (See 1.4). So we can estimate the grade of expression by the materials state's similarity to the ground data. As expected, the models, produced an output which implied in principle the same state of material, as it was present in the ground data.

All the models as far as the *learned*, applied *what* they learned. There was no escape from the delivered state of materials. As an example here different results referring to the same starting-file. (Midi files mixed down as as mp3).[35]

- Toy song from ground data  
[https://github.com/fredrikzellerR/bacc/blob/main/wav/3\\_ModArp\\_2.mp3](https://github.com/fredrikzellerR/bacc/blob/main/wav/3_ModArp_2.mp3)
- New song made of the first 16 beats of the Toy song.
  - LSTM crossentropy loss.  
[https://github.com/fredrikzellerR/bacc/blob/main/wav/3\\_ModArp\\_2\\_5100.mp3](https://github.com/fredrikzellerR/bacc/blob/main/wav/3_ModArp_2_5100.mp3)
  - RNN Trivial loss.  
[https://github.com/fredrikzellerR/bacc/blob/main/wav/3\\_ModArp\\_2\\_620.mp3](https://github.com/fredrikzellerR/bacc/blob/main/wav/3_ModArp_2_620.mp3)
  - GRU crossentropy loss.  
[https://github.com/fredrikzellerR/bacc/blob/main/wav/3\\_ModArp\\_2\\_710.mp3](https://github.com/fredrikzellerR/bacc/blob/main/wav/3_ModArp_2_710.mp3)
  - GL  $\Lambda = 0.0001$  crossentropy loss.  
[https://github.com/fredrikzellerR/bacc/blob/main/wav/3\\_ModArp\\_2\\_300001.mp3](https://github.com/fredrikzellerR/bacc/blob/main/wav/3_ModArp_2_300001.mp3)
  - GL extended  $\Lambda = 0.0001$  BCE.  
[https://github.com/fredrikzellerR/bacc/blob/main/wav/3\\_ModArp\\_2\\_20128256.mp3](https://github.com/fredrikzellerR/bacc/blob/main/wav/3_ModArp_2_20128256.mp3)



---

As a conclusion, the output of the net is exactly as expressive as the ground data. At the end this is no surprise, because we didn't give other chance to the net. We can speculate that providing more and different training data, would not change this. The net would reproduce equally more or less the state of material, which is represented in the - now extended - ground data. Extending, multiplying and differentiating the training data will not change the basic behaviour of the net. This leads us back to the question: "Is AI able to replace the author of an piece of art-music?".(1.5)

The answer is probably: It would be possible, but only if the AI would achieve a state, in which it produces a kind of *self-consciousness*. The composing AI would have to know, *what* it is doing. On the contrary, we assume that for a neuronal net it's stochastic type of calculation is highly abstract, not bound on a specific domain - in our case: music. Formulated in a provocative manner: The *same* calculation, which generates pieces of music, could fit, to decide how to arrange a vegetable shelf in a supermarket.<sup>1</sup>

During the creation process, an author (artist) at any time is able, to make a step aside, and observe his work from 'outside'. Comprehending what he has done so far, from this point of view, is looking at himself. This opens up the possibility to give at the whole for example a new direction, another twist.

This is the home of the, in most contexts depreciated, word 'creativity'. It would mean, reorganizing a domain with the help of thoughts, structures which are *not* intrinsic to that domain.

The basic question is: Is self-consciousness a question of quantity? Could an AI achieve self-consciousness if it would only been fed with enough data, and if it would dispose about just enough neurons? Is there then a tipping point, where stochastic calculus flips into something else ?<sup>2</sup>

---

<sup>1</sup>This is obvious if we take in count the fact, that there is a wide use of so called *pre-trained* models. The 'new black': Transformer networks.

<sup>2</sup>Very actual questions, considering the current discussion around chatGPT. A conversation around March 2023:

"Human: Hello, who are you?

AI: Hello! I am ChatGPT, a language model developed by OpenAI. I am designed to respond to your questions and generate human-like text based on my training on vast amounts of data. How can I assist you today?

Human: Do you have self-consciousness?

AI: ..., I do not have self-consciousness or a sense of self-awareness like humans do. ... My responses are based on my programming and the data I have been trained on, and I do not have personal opinions or preferences.

Human: Please define self-consciousness.

AI: Self-consciousness is the awareness of one's own existence, thoughts, feelings, and experiences as a distinct and separate entity from the external environment and other individuals. It involves the ability to reflect on oneself, recognize one's own identity, and understand one's place in the world..."[34]

Let's invent two speculative counterexamples:

### Beethoven 9th Symphony

Given that we move the AI back to 1822, we feed the AI with all the music up to this time. The training material covers instrumental works in all type of orchestration, as well as numerous vocal works. Setting the focus on the choir (and soloists) in the last movement of Beethoven's 9th symphony, we can assume, that the AI, on the demand to compose a new symphony, would probably built in vocal passages. *But* we would notice, that this happens statistically distributed all over the duration of the piece. We would 'feel' the statistic. Whereas the achievement of Beethoven is, to put these vocal parts concentrated, in a manner of statement, which goes perfectly with the content of the lyrics. Furthermore we notice the immanent influence of vocal phrasing on the orchestra: The 'introduction'<sup>3</sup> in the 4th movement by cello and double-basses is made in a style of an *recitative*. The choir-type reflects back on the rest of the musical material. We doubt heavily that an AI would have done this double transfer of ideas.

### Nikolaus Kopernikus

Back to 1500, well known the theory of epicycles. We feed the AI with all the planetary and stellar observations at the time. Then it happens that we discover a new object in the sky. We ask the AI, to calculate / complete the orbit of that object. Very likely the AI would give us an answer, containing more or less complicated epicycles.

Also very likely: The AI would never assume, that epicycles are something strange or complicated, and never would consider that putting the sun into the centre, gives a simpler and therefore better explanation.

Both examples show that for 'higher' ideas (we implicate that in art-music) it's necessary to step outside of the construct of ideas, take a higher point of view, to finally reconstruct the whole.

So what for the AI?

In short: Its a tool, nothing more.

This is exactly how AI can be useful now a days. The power of AI is its ability to manage large data volumes and to deal with patterns. The AI may evaluate sky surveys, or collision results in a particle accelerator, as well as judge a lot of

---

<sup>3</sup>4th movement. Presto. Bar 8 ff.

---

photos in medical applications.

As a composer you may use an AI if you want to dispose quickly of some materials. You may train a neuronal net, which delivers a bunch of proposals, always in the sense of the training material. It may happen that you detect in there some new aspects, you didn't see 'by hand', because the ground data itself is too extensive, or somehow too complicated. But you are aware at any time, that, what the AI does are basically *variants* of your ground data.





## Appendix A

# Toy song in B-flat major

The musical score is written for piano in B-flat major, 4/4 time. It consists of seven systems of staves, each with a treble and bass clef. The key signature has two flats (B-flat and E-flat). The score includes various musical notations such as chords, single notes, and rests. Measure numbers 9, 17, 25, 33, 41, and 49 are indicated at the start of their respective systems. At measure 25, there is a 'Modulation -> C-minor' annotation. At measure 41, there is a 'Back Modulation' annotation. At measure 49, there is a 'DDv' annotation. The score ends with the word 'etc.' in the final measure. Below the final system, there are additional annotations: '4 6' under the first measure, '3 5' under the second measure, and 'B-major' under the third measure.

9

17

25 Modulation -> C-minor

33

41 Back Modulation

49 DDv etc.

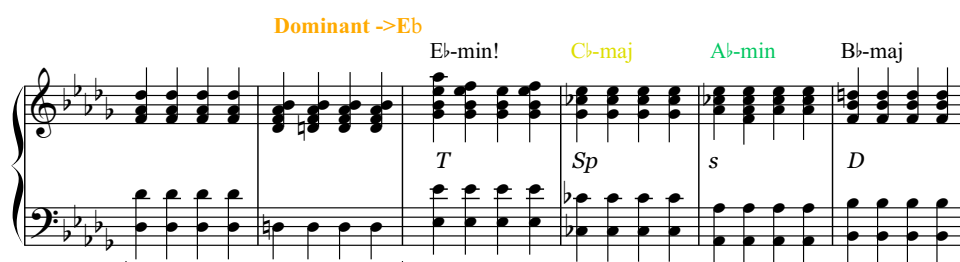
4 6 3 5 B-major



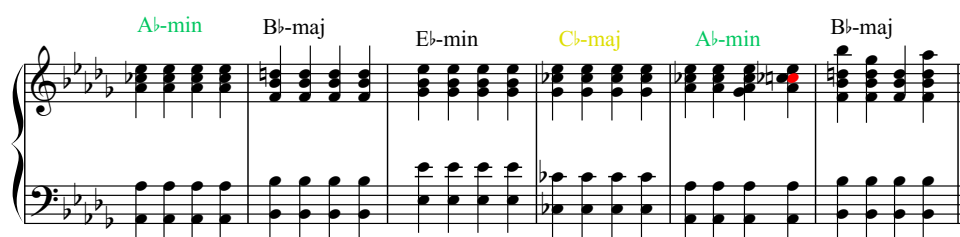
## Appendix B

# Result Db-Major modulated to Eb-minor

1ModArp1, GL Lamda 0.0001



This chromatic progression is supposed to be used for modulation backward! ----- but follows correctly a cadence in Eb-minor obviously the model thought to modulate back from Db-Minor, which would be *Sp*







## Appendix C

# Analysis of Bb-Major

-1ModArpo

BCE extended

F# ? g# ?

RNN BCE

B-mi

ca. Db

≈E-mi

≈A-ma

≈F#-mi

D-ma

B-mi

--> F#

C#-ma

F#-mi

D-ma

(G-ma)

LSTM Trivial-loss

≈Bb-mi

≈D-ma

2 GL model BCE

D $\sharp$ -mi (subdominant-parallel in B-ma)  
enharmonic change to E $\flat$ -mi...

F-ma ... what is interpreted  
as subdominant in B-mi

TP S D

G $\flat$ -ma E $\flat$ -mi ... regular cadence in B $\flat$ -mi

GL model Cross-entropy

B-"void"

E-mi  
E-ma

C-ma

A-mi B-ma as dominant E-mi (void) C-ma as Subdominant in G D-ma G-ma



# Bibliography

- [1] Christian Gumbsch, Martin V. Butz (2021). *Sparsely Changing Latent States for Prediction and Planning in Partially Observable Domains*.
- [2] ibid. Page 2.
- [3] ibid. Page 3.
- [4] ibid. Page 4.
- [5] Christian Gumbsch, martius-lab.(2021) <https://github.com/martius-lab/GateLORD-paper>
- [6] Christian Gumbsch, Martin V. Butz (2021). *Supplementary Material for: Sparsely Changing Latent States for Prediction and Planning in Partially Observable Domains* Page 24. Figure 16
- [7] Walter Benjamin (1935). *Das Kunstwerk im Zeitalter seiner Technischen Reproduzierbarkeit*. Chapter II.
- [8] Douglas Eck ,Jürgen Schmidhuber. (2002) *A First Look at Music Composition using LSTM Recurrent Neural Networks*. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, vol. 103.
- [9] ibid. Page 3. Chapter 2.1. LSTM Architecture.
- [10] Piyush Kumar Arya, Pranshu Kukreti and Nilabh Jha. *Music Generation Using LSTM and Its Comparison with Traditional Method* Page 4. 4. Solution to traditional method.
- [11] Midi Association. Official MIDI Specifications. (2023) <https://www.midi.org/specifications>
- [12] TensorFlow. Generate music with an RNN.(2023) [https://www.tensorflow.org/tutorials/audio/music\\_generation](https://www.tensorflow.org/tutorials/audio/music_generation)
- [13] Juan Luis García López. chord suggerter.(2020) <https://github.com/huanlui>

- [14] Mateusz Modrzejewski, Mateusz Dorobek, Przemysław Rokita. *Application of Deep Neural Networks to Music Composition Based on MIDI Datasets and Graphical Representation*
- [15] Cheng-Zhi, Anna Huang, Ashish Vaswani, Jakob Uszkoreit, Noam Shazeer, Ian Simon, Curtis Hawthorne, Andrew M., Dai Matthew, D. Hoffman, Monica Dinulescu, Douglas Eck. *MUSIC TRANSFORMER: GENERATING MUSIC WITH LONG-TERM STRUCTURE*. Page 11. A.1 SERIALIZED INSTRUMENT/TIME GRID
- [16] ibid. Page 8 4.2.3 HUMAN EVALUATIONS
- [17] Pytorch. PyTorch Contributors. CrossEntropyLoss. (2023) <https://pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html>
- [18] Youtube. See You Again - AI Composed Relaxing Music by AIVA. (2023) <https://www.youtube.com/watch?v=FG3ZJ5WHzxM> with Aiva Technologies SARL (2016-2023) <https://www.aiva.ai/>
- [19] Geekflare.(2023) <https://geekflare.com/de/ai-art-generators/>
- [20] Anton Webern. (1912) *Schönbergs Musik*. Essay.
- [21] Jean-Philippe Rameau.(1722) *Traité de l'harmonie*. Paris
- [22] Hugo Riemann.(1. Auflage 1880) *Handbuch der Harmonielehre*. Breitkopf & Härtel, Leipzig 1929
- [23] Arnold Schönberg.(2. Auflage 1922) *Harmonielehre*. Universal Edition Wien 1911.
- [24] Diether de la Motte. (1976) *Harmonielehre*. Bärenreiter, Kassel, und Deutscher Taschenbuch Verlag, München.
- [25] Zeitschrift der Gesellschaft für Musiktheorie (ZGMTH). Georg Olms Verlag, Hildesheim
- [26] Hector Berlioz.(1843–1844) *Grand Traité d'instrumentation et d'orchestration modernes*.
- [27] Helmut Lachenmann.(5. März 1996) *Klangtypen der Neuen Musik*. From „Musik als existentielle Erfahrung“ Breitkopf & Härtel; 2., Edition
- [28] Fred Lehndahl , Ray Jackendorf.(1983) *Generative Theory of Tonal Music*.
- [29] Heinrich Schenker.(1906) *Neue musikalische Theorien und Phantasien*. Stuttgart, Berlin

- [30] Dimitris Rafailidis, Alexandros Nanopoulos, Emilios Cambouropoulos, Yannis Manolopoulos. DETECTION OF STREAM SEGMENTS IN SYMBOLIC MUSICAL DATA. Abstract.
- [31] Mathieu Giraud, Richard Groult, Florence Levé. *Computational Analysis of Musical Form*. 5.2.1 Theories of Form.
- [32] Louis Bigo , Moreno Andreatta. *Topological Structures in Computer-Aided Music Analysis*
- [33] For example: *What Makes This Song Great?* Ep.112 TEARS FOR FEARS. <https://www.youtube.com/watch?v=6bU4R04fH4U>
- [34] OpenAI. Introducing ChatGPT. (2023) <https://openai.com/blog/chatgpt>
- [35] Relevant code in Python here: GitHub. fredrikzellerR/bacc (2023) <https://github.com/fredrikzellerR/bacc/tree/main/code>