HfM Detmold
HOCHSCHULE FÜR MUSIK

Submitted by
**Alexander Leemhuis**

Submitted at
**Center of Music and Film Informatics**

Supervisor and First Examiner
**Prof. Dr. Aristotelis Hadjakos**

Second Examiner
**Simon Waloschek**

October 2019

# Musicologically Informed Artificial Bach Chorale Harmonization

Bachelor Thesis

to obtain the academic degree of

Bachelor of Music

in the subject

Musikinformatik

**Detmold University of Music**
Neustadt 22
32756 Detmold
Germany

www.hfm-detmold.de

# Abstract

Chorales by Baroque composer Johann Sebastian Bach have been used to teach music students the basics of music theory for decades. In recent years, this topic has also become an important field of research for computer scientists: Several approaches have been developed to automatically write music in the style of Bach. In this research, a neural network architecture based on musicological literature and the actual human composition process is proposed. Online evaluations with expert listeners show that the generated chorales highly resemble Bach's harmonization style, additional musical analyses of the chorales show possibilities for further improvement.

# Zusammenfassung

Choräle des Barockkomponisten Johann Sebastian Bach werden seit Jahrzehnten dazu genutzt, Musikstudierenden musiktheoretische Grundlagen zu vermitteln. In den letzten Jahren ist dieses Thema auch für Datenwissenschaftler/-innen zu einem wichtigen Forschungsgebiet geworden: Um automatisiert Musik im Stil von Bach zu schreiben, wurden mehrere Ansätze entwickelt. In dieser Arbeit wird eine neuronale Netzwerkarchitektur vorgestellt, die auf musikwissenschaftlicher Literatur und dem menschlichen Kompositionsprozess basiert. Online-Evaluationen mit Experten zeigen, dass die erzeugten Choräle dem Harmonisierungsstil von Bach sehr ähnlich sind, zusätzliche musikalische Analysen der Choräle zeigen Verbesserungsmöglichkeiten auf.

# Statutory Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own. The submitted written copies and the digital version are consistent with each other in contents.

Detmold, October 8, 2019

Alexander Leemhuis

# Contents

# 1 Introduction

For many musicians, chorales by late Baroque composer Johann Sebastian Bach are the epitome of four-part vocal works [4, p. 9]. Bach chorales are a popular object of study, an integral part of Western music education, and belong to the repertoire of many choirs. In those chorales, several important aspects of Western music theory are united: counterpoint, harmony, Baroque stylistic conventions. Thanks to the high extent of uniformity in his chorales, the rules that make the chorales easy to grasp, and the large quantity of digitally available chorales, the works of Bach are also an interesting topic for computational music generation. Especially automatic harmonization, or more precisely producing a four-part chorale given the soprano part, has been an important field of research for decades.

Although various musicological textbooks deal with the right procedure when writing Bach chorales, the therein proposed methods have not been used to design a neural network architecture. Therefore, in this research, a Convolutional Neural Network (CNN) architecture in accordance with the actual human composition process described in common music theory literature is developed. The proposed method has already successfully undergone a peer review process and has been accepted by the scientific community [19].

The generated chorales are evaluated both by a musical analysis and an expert evaluation addressing music majors and musicologists. The online evaluation shows that the results of the network are convincing: In serveral cases, the original Bach chorale could not be correctly identified in direct comparison to the generated harmonization of the same soprano part. Some generated chorales were even believed to be the work of Bach. In-depth music-theoretical analyses of the generation results give insights into the capabilities of the trained network model. They point out limitations as well as room for further improvement and allow hypotheses about the composing style of J.S. Bach.

Although writing a four-part chorale requires a lot of technical knowledge, “abiding by the rules” is undisputedly not enough to produce a good chorale. Previous algorithmic approaches have indeed shown that manually gathering up all discoverable rules that seem to constitute Bach chorales does not lead to results in the style of the Baroque composer [5]. The voice leading rules known to us are strict but still seem to allow room for making aesthetic pieces of art. Complex allusions, word-music relationship, and the courage to break voice leading rules if it serves some artistic purpose result from Bach’s musical listening habits that manifest themselves in his chorales. This is something that can’t yet be transformed into computational rules. Music students can tell from their first semester music theory lesson: A housework chorale may not be wrong, but it is surely not comparable to the master’s works.

J.S. Bach himself never made a detailed remark about the music theory behind his compositional style [4, p. 9]. This work tries to do a step towards finding out if the subtleties of Johann Sebastian Bach’s abilities are lost - or hidden but discoverable in his works. If one can’t fully reconstruct them from learning the rules - a self-learning artifical intelligence may hint at the features that we overlook and reapply them to new Bach style works.

# 2

# Background

## 2.1 Artificial Intelligence and Deep Learning

The following explanations are based on the textbook “Deep Learning” by Goodfellow, Bengio and Courville [7].

To create computer programs that solve tasks in a human-like manner, several approaches have been developed. In order to be accessible for computers, information about the world had to be remodelled in a formal way. Therefore, in early days of *artificial intelligence* (AI), real-world notions and rules were hard-coded by humans. Often, those attempts remained unsuccessful because the material turned out to be too complex and the programmed rules could not accurately describe the environment.

*Machine learning* is a collective term for artificial intelligence approaches where instead of using hard-coded rules, the computer algorithm can “acquire their own knowledge, by extracting patterns from raw data” [7, p. 2]. This renders these approaches suitable for real-life applications. *Deep learning* is considered a machine learning subdiscipline (see Fig. 2.1) where computers resolve problems by breaking down complex tasks into individual parts and subconcepts. Through several layers, a neural network is able to conceptualize the environment and develop a hierarchy of subconcepts.

A feedfoward neural network, the most common example in the field of deep learning, corresponds to “a mathematical function mapping some set of input values to output values” [7, p. 5].

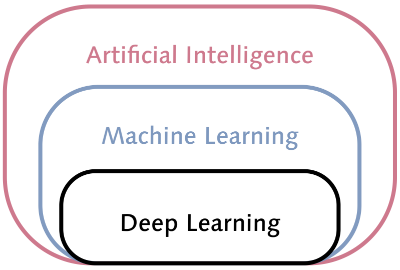



**Figure 2.1:** Schematic representation of the relationship between artificial intelligence and the sub-disciplines.

The approach described in this research uses a deep neural network with a particular hidden layer structure, so it can develop a hierarchy of concepts to grasp more abstract ideas and solve the harmonization task. A symbolic representation of the original soprano part and additional score information serves as an input matrix, the full original Bach work as an output matrix for the network. Using algorithms like back-propagation [24] and the Adam optimizer [16], these matrices are used to transform the layers of the network in order to minimize the deviation from the "correct" chorale during the training process. This way, the algorithm "learns" how to write full four-part chorales based on the training melodies. Later, the learned abilities can be examined and evaluated using new melodies as unseen data.

## 2.2 Compositional Technique of Bach Chorales

The Bach chorales, on which this research concentrates, play an important role in Western music education. Since those late Baroque works follow an aesthetic permeated by both a harmonic and a counterpoint perspective on music [11, p. 51], they serve as a suitable object of study to convey the basics of Western music theory. Writing four-part chorales in the style of Bach is thus covered in many different musical fields of study and a common excercise in music theory lessons. In order to

- understand the way this harmonization task is carried out musicologically and
- to develop an approach to computationally process the chorales,

one has to take a closer look at the inner structure of those pieces.

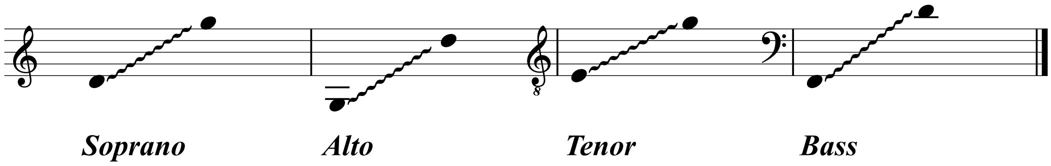



**Figure 2.2:** Vocal ranges of the parts as used by J.S. Bach [4, p. 62].

## Music-theoretical Basics

The chorales usually consist of four independent parts, the *soprano*, *alto*, *tenor* and *bass* part. All groups stick to specific pitch ranges, see Fig. 2.2.

Horizontally, these pieces progress mainly in quarter notes; the smallest note value that Bach uses is a semiquaver. Many melodies and texts that the chorales are based upon date from the 16th century: J.S. Bach rarely uses contemporary Baroque melodies, but instead modifies traditional tunes and adapts them to fit late Baroque stylistic conventions [4, p. 222]. The line endings of those texts correspond to musical phrase endings - the standardized formulaic expressions found here are called cadences and can be both interpreted as chord progressions and simultaneous sounding of independently moving vocal parts. They are usually marked with a fermata.

Concerning the musical content of the chorale and the movement of the parts there are numerous rules and restrictions like e.g. the well-known prohibition of parallel fifths and octaves [4, p. 80] and the resolution of dominant seventh chords.

## Musicologically Informed Harmonization

To guide students engaged with this tasks to cope with these voice leading rules, a lot of musicological works and instruction textbooks have been written over the centuries.

A fundamental recommendation from contemporaries of the Baroque era as well as modern experts is to start the harmonization process with the lowest voice [4, p. 237, 15, p. 75]. The bass part of Bach chorales is not only considered one of four equitable voices but also an indicator of the tonal skeleton. It contains information about the harmonic framework and determines the fundamental structure of the piece.

As a second step, the middle voices are created. G.Ph. Telemann argues that after the bass part was written, the alto part should be written before the tenor part so that the closest possible voicing can be accomplished [26, p. 183]. In some places, there can be very small leeway for the alto and the tenor part. Oftentimes, there is only one solution for a valid choice of alto and tenor notes, so they can be very plain and motionless, only blending into the harmonic progression [4, p. 255]. It is possible that in certain situations no solution can be found in accordance with voice leading rules. In those cases, it's common to go back and revise certain notes and develop an alternative approach.

As explained before, the cadences, typically marked by fermatas, should be prepared in advance to define the harmonic structure more clearly [4, p. 159].

In summary, the following harmonization strategies can be derived from musicological expert knowledge:

- Generate the bass part first given the soprano part
- Support close voicings by choosing the tenor notes after the alto notes
- Give enough context to prepare the cadences at the ends of phrases
- Allow changes to previously generated notes to try alternative solutions

These guidelines will later be used to derive a neural network architecture to carry out the harmonization task computationally.

# 3 Related Work

Several approaches have been developed in recent years to not only automatically generate symbolic music in general but also to specifically harmonize melodies of Johann Sebastian Bach.

The first approach was a so called expert system from 1986. It used a set of 270 hand-engineered rules, e.g. constraints for dominant resolutions [5]. The results were of state-of-the-art quality at that time, but the style of the resulting chorales was not Bach's, except for some typical phrases.

Since the late 1980s, various approaches based on neural networks were developed. Early neural network algorithms that could be fully implemented belonged to a second wave of neural network research known as "connectionism" [7, p. 13]. Todd [27] and Mozer and Soukup [21] show that a network can be able to learn and recreate monophonic melodies exposed to the network. The later approach HARMONET [9] produces full four-part chorales. It harmonizes melodies by reducing the chorale to a harmonic skeleton first and later filling it with quaver ornamentations.

In 2002, for the first time, long short-term memory cells (LSTM) [10] were used for music generation. LSTM networks have special components that enable the network to memorize values and are thus well-suited to model sequences [7, p. 18]. Eck and Schmidhuber show that recurrent neural networks based on LSTM cells are suitable to recreate the structure of blues music [6].

Since the following decade, numerous statistical models were developed. In [1], Allan and Williams present an algorithm that learns from original Bach chorales and creates own harmonizations using Hidden Markov models and a probabilistic framework. First, a harmonic progression in quarter notes is developed. As a second step, a different network creates quaver ornamentations. Although the results feature some Baroque characteristics, like contrary motion between different parts and harmonic progressions, there are unsuitable jumps in the individual parts and the time resolution is limited to quaver notes which is not sufficient to to represent Bach chorales entirely [4, p. 117].

Also, Boltzmann machines were used to analyse [2] and recreate [18] structure of polyphonic music. In [25], Suzuki and Kitahara used Bayesian networks to generate chorales based on the soprano part and examine the advantages and disadvantages of representing chords with different voicings using the same symbol. They showed that it is favorable to generate the bass line of a chorale first and that modelling the chorale with individual notes rather than with chord symbols leads to smoother parts. The advantages of generating the bass line first in computer systems that generate chorales were also examined in [28].

In 2016, the so-called BachBot [20] and the DeepBach approach [8] were developed in parallel. They both use LSTM cells again and take metadata like the metrical positions of notes and fermatas into account but disregard additional score information like the time signature and the key signature. Both approaches were each evaluated with a discrimination test. In [20], the participants had to find the original composition in direct comparison to the generated chorale. It was shown that in the harmonization case (alto, tenor and bass part were generated) the participants were more likely to find the original chorale than in those cases where the full chorale was generated or only one of the parts was filled in.

A recent contribution to this field of research is the Bach Doodle [13] developed by technology company Google. Based on Coconet, a deep convolutional model that is able to repopulate incomplete scores or harmonize melodies [12], it allows users of a search engine to harmonize own melodies with an interactive sheet music interface online in the webbrowser.

In this research, a discrimination test is performed, similar to BachBot [20]. Also, taking into account findings by Suzuki and Kitahara [25] and Whorley et al. [28], the bass line is generated in advance. But for the first time, in contrast to previous work, this work establishes the entire network architeture based on musicological insight.

# 4

# Data Processing

The first step towards this music generation approach is to determine the way in which symbolic music is to be represented. Bach's compositions must be transformed into machine-readable data structures so that they are accessible to a neural network.

## 4.1 Selection and Preprocessing

To retrieve and process score data, the music21 framework [3] for Python was used. Music21 is a Python library that provides extensive possibilities to process symbolic music and a corpus with various pieces by composers such as Johann Sebastian Bach in MusicXML format. PyTorch [23] was used as machine learning framework.

Of all available chorales, only those with exactly the four voices *soprano*, *alto*, *tenor* and *bass* were selected. Pieces with fewer parts and pieces with orchestra accompaniment were ignored. The remaining 348 chorales were split randomly: 95% of the chorales (331 pieces) were used for the training process and 5% (17 pieces) were used to tune the hyperparameters of the network and to evaluate the results.

Because J.S. Bach sometimes uses the same soprano part to compose several chorales, some melodies may be present in the training as well as the test dataset. The harmonizations in such cases are still different, so, in accordance with similar generative systems [8, 20], this circumstance is not further taken into account. Since grace notes are mostly used for suspended notes, they could be left out. Repeat marks were ignored since the repeated music contains no additional harmonic information.

## 4.2 Data Augmentation

The selected chorales are not uniformly distributed over the key and time signatures, J.S. Bach seems to prefer keys with less accidentals like C major and G major and their relative minor keys. 89% of all processed chorales have a $\frac{4}{4}$ time signature, see Fig. 4.1.

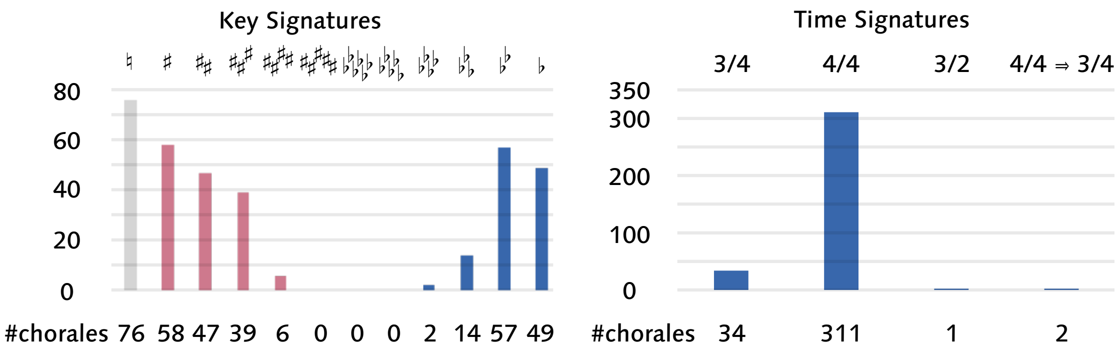



**Figure 4.1:** The distribution of the 348 chorales over the key and the time signature. The most common key signature has no sharps and no flats which corresponds to the keys of C major or A minor. The most common time signature is $\frac{4}{4}$. Two chorales start with a $\frac{4}{4}$ time signature and later change to $\frac{3}{4}$.

The pieces are not transposed to the key of C major or A minor, as it was implemented in prior work [20]. Instead, to augment the dataset, they are transposed to different keys limited in such way that no part exceeds the vocal ambitus used by Bach. This has following benefits:

- The resulting chorales always remain singable.
- A voicing at a certain scale degree can now also be applicable to chords on other scale degrees. E.g., tonic voicings in the key of F major benefit subdominant chords in the key of C major since it's the same triad. Transposing all pieces to the same key would render this impossible.

The augmented dataset consisted of 2583 chorales. Rhythmic augmentation, i.e. accelerating or slowing the note values by a certain factor, was not implemented because

- it would distort the positions of the notes within the measures and
- it would distort the tempo of harmonic progression.

## 4.3 Format

In order to feed the chorales into the harmonization algorithm, the music pieces had to be transferred into a machine-readable data format. As described in Sec. 2.2, J.S. Bach does not use note values shorter than a semiquaver in his chorales, so a time grid of semiquavers has been used without disregarding any temporal information. For each semiquaver step, several musical attributes were computed both for each of the parts soprano, alto, tenor, and bass and also globally for the piece.

### Part Representation

For each voice, the following pieces of information were computed:

- If a new note starts at the given time step, a special **pitch value** is encoded.
- **Rests** are treated like notes, encoded as a pitch beyond the pitch ambitus.
- In cases where a note is tied over, a **continuation** flag is set.

Later, these pieces of information were transformed into a sequence of one-hot encoded vectors: For every time step and each possible activity of a part (notes/rest/continue), a binary value was calculated, see Fig. 4.2. See Tab. 4.1 for an overview of the part encoding.

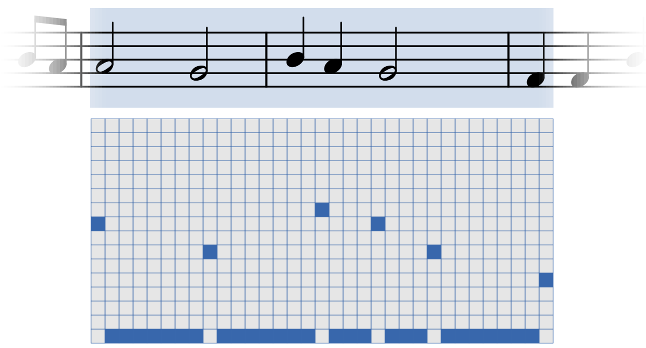


**Figure 4.2:** Visualization of the data scheme. It shows 33 one-hot encoded semiquavers of the soprano part, the boxes on the bottom correspond to the continuation flag.

| Data Soprano | Possible Values |
|---|---|
| ... | |
| Pitch C4 | $\{0, 1\}$ |
| Pitch C♯4 | $\{0, 1\}$ |
| Pitch D4 | $\{0, 1\}$ |
| ... | |
| Rest | $\{0, 1\}$ |
| Continued | $\{0, 1\}$ |

| Data Alto |
|---|
| ... |
| Pitch C4 |
| Pitch C♯4 |
| Pitch D4 |
| ... |
| Rest |
| Continued |

. . .

**Table 4.1:** The encoding of the symbolic music. Per part, pitch heights, rests, and ties are computed.

## Metadata

Similar to prior work by Liang [20], global score information was also added to provide more information about the music. Involving all the aspects of the chorale seen in Tab. 4.2 in the data scheme aims at representing the piece in its entirety:

- Fermatas in the chorale denote phrase endings and cadences. Therefore, it's crucial to encode the **presence or absence of fermatas** as these special chord progressions have to be prepared earlier as suggested by Daniel in [4, p. 159].
- The **key signature** provides information about the tonal context of the passage and is encoded as number of sharps.
- Fundamental differences in effect of the music are often caused by the arrangement of pulses within the measure. The **time signature** adds information about these inner structures of the measures.
- Different beats within a measure can be stressed and unstressed depending on the time signature. To account for that, the **position within the current measure** was added.
- At Bach's time, equal tempering was not yet established, in the available tuning systems, different keys had their unique characteristics [4, p. 11]. After the augmentation of the data set, the original key can't be unambiguously reconstructed. To maintain a possible "key character", a **pitch offset** was added after the data augmentation, referring to the distance to the original fundamental tone.

| Metadata | Possible Values |
|---|---|
| Fermata | $\{0, 1\}$ |
| Time Signature 3/4 | $\{0, 1\}$ |
| Time Signature 4/4 | $\{0, 1\}$ |
| Time Signature 3/2 | $\{0, 1\}$ |
| Time Position in Quarter Notes | $\{0, 0.25, 0.5 \ldots\}$ |
| Pitch Offset in Semitones | $\mathbb{Z}$ |
| Key Signature ♮ | $\{0, 1\}$ |
| Key Signature ♯ | $\{0, 1\}$ |
| ... | |
| Key Signature ♭♭ | $\{0, 1\}$ |
| Key Signature ♭ | $\{0, 1\}$ |

**Table 4.2:** An overview of additional information about the score that is included.

The resulting semiquaver grid values contained four individual part data arrays and one global array (see Tab. 4.2). With these pieces of information it's possible to fully represent all important aspects of the chorale and feed it into the network.

# 5

# Network Architecture

Following the advice of musicological literature on the human composition process, the harmonization workflow should be divided into two steps. Knowing only the soprano part, the bass part (and thus the harmonic framework) should be created first. Once the entire bass part has been written, the middle parts, i.e. the alto and tenor notes, are produced in a second step. This is the basis for the architecture of the network.

## 5.1 Generation Process

To implement the above, the computational generation process was split into three different consecutive networks:

- A first network generates the bass line.
- A second and a third network generate the middle part notes alternatingly, tenor note after alto note for each time step.

### Bass Generation

The first network takes a frame of the soprano part and the metadata as well as prior bass notes as an input. The frame size has to be sufficient to prepare cadences as suggested by Daniel [4, p. 159], so the frame size was added to the set of hyperparameters (see Sec. 5.2) and finally set to 8 quarter notes for future and prior notes each.

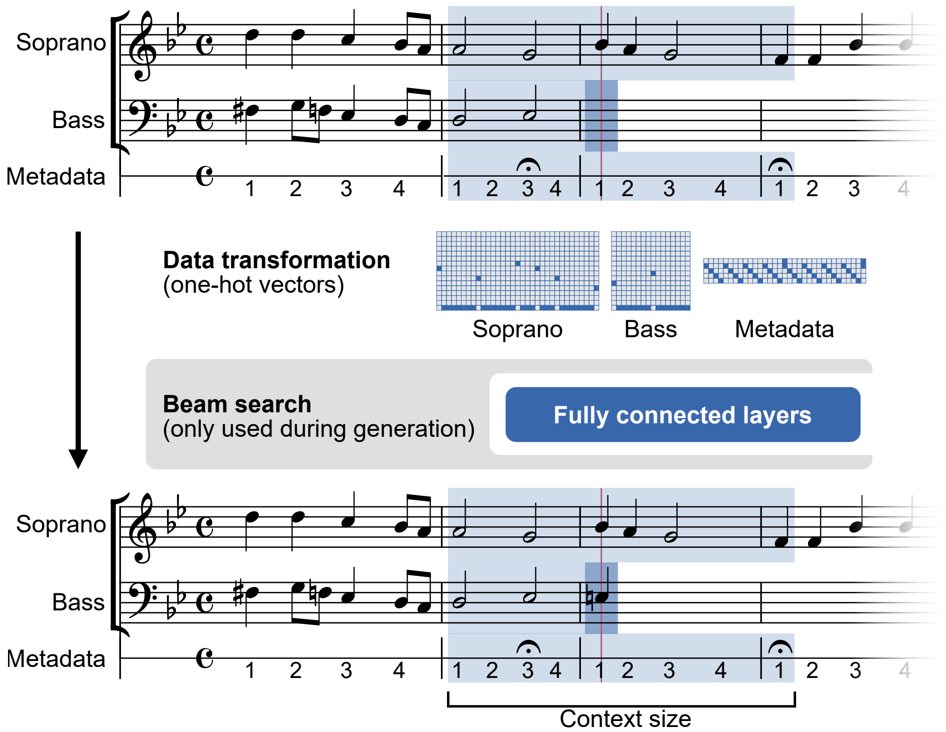



**Figure 5.1:** Scheme of the bass part generation. The one-hot encoded data is fed into several fully connected layers to generate the output for a single time step. Only one hidden layer was used, because deeper networks didn't increase the performance. Afterwards, the context window is shifted by one step into the future. The context size shown in blue is deliberately reduced compared to the actual implementation to enhance readability.

Eight quarter notes equal 32 semiquaver time steps, so the bass event $b_i$ depends on:

- the soprano part in a local context of ±32 time steps $s_{i-32:i+32}$
- the metadata in a local context of the same size $m_{i-32:i+32}$
- the 32 previous bass events $b_{i-32:i-1}$

The probability model for predicting $b_i$ is:

$$p(b_i|s_{i-32:i+32}, m_{i-32:i+32}, b_{i-32:i-1})$$

See Fig. 5.1 for a graphical representation of the bass generation process.

### Alto and Tenor Generation

For the generation of the middle parts, two alternating networks were used. As before, they take the additional score information and the soprano part but also the entire bass line and the prior middle part notes as an input. Therefore, the alto prediction $a_i$ depends on:

- the soprano and metadata context $s_{i-32:i+32}$, $m_{i-32:i+32}$ as above
- the prior and future bass events $b_{i-32:i+32}$ generated in the first step
- all prior alto $a_{i-32:i-1}$ and tenor events $t_{i-32:i-1}$

The resulting probability model is:

$$p(a_i|s_{i-32:i+32}, m_{i-32:i+32}, b_{i-32:i+32}, a_{i-32:i-1}, t_{i-32:i-1})$$

The tenor note also depends on the alto note generated in the current time step:

$$p(t_i|s_{i-32:i+32}, m_{i-32:i+32}, b_{i-32:i+32}, a_{i-32:i}, t_{i-32:i-1})$$

## 5.2 Implementation Details

To find the optimal hyperparameters for the three networks, random searches and grid searches in immediate surroundings of best random search results were performed. The architecture finally used had the following properties:

- a learning rate of $0.5 \times 10^{-3}$ with a decay of 0.99 every 30 epochs
- one single hidden layer with the size of 650
- a context size of 32 semiquaver time steps
- a dropout of 0.5 before the hidden layer and the output layer

Input and output layer dimensions are defined by the individual pitch range of each part. The output layers use softmax nonlinearities to create probability distributions, all other layers use SELUs [17]. Adam [16] was chosen as learning rate optimization algorithm and, as implemented in previous approaches [20], cross-entropy was chosen as loss function.

## 5.3 Beam Search

A human composer has always the option to adjust and rework certain parts of their work. In a position in which no continuation of the chorale part can be found that complies with the voice leading rules, the composer can always revise previous notes.

In generative computer systems, that doesn't necessarily apply. When determining each note of a sequence from start to finish based on local probabilities only, the possibility for revision is not inherent. In those "greedy" cases, the total probability - the product of the predicted local probabilities $p(b_i|\cdot)$, $p(a_i|\cdot)$ and $p(t_i|\cdot)$ for each time step $i$ depending on the local context - is not always the optimum.

As suggested in related work in Bach chorale harmonization [20], a beam search [22, p. 195] algorithm was implemented to maximize this total probability

$$P = \prod_{i=0}^{N} p(b_i|\cdot)\, p(a_i|\cdot)\, p(t_i|\cdot).$$

The beam search algorithm enables the neural network to maintain multiple high-probability paths of the search tree while generating the sequence. The number of candidates is limited to minimize runtime and memory requirements. Those candidates can then serve as alternative solutions that can be further expanded when one solution leads to a dead end with a low total probability. Up to now, this has not been implemented and evaluated in similar research.

See Fig. 5.2 for a graphical example of the beam search algorithm implemented in the bass part generation. Alto and tenor parts are generated similarly.

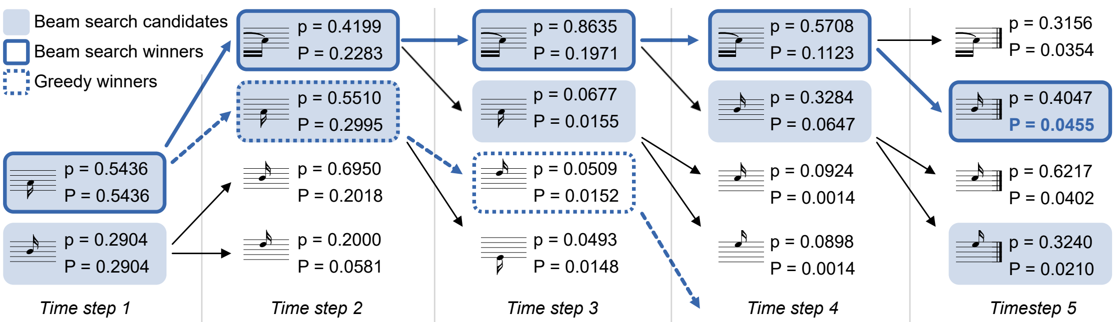
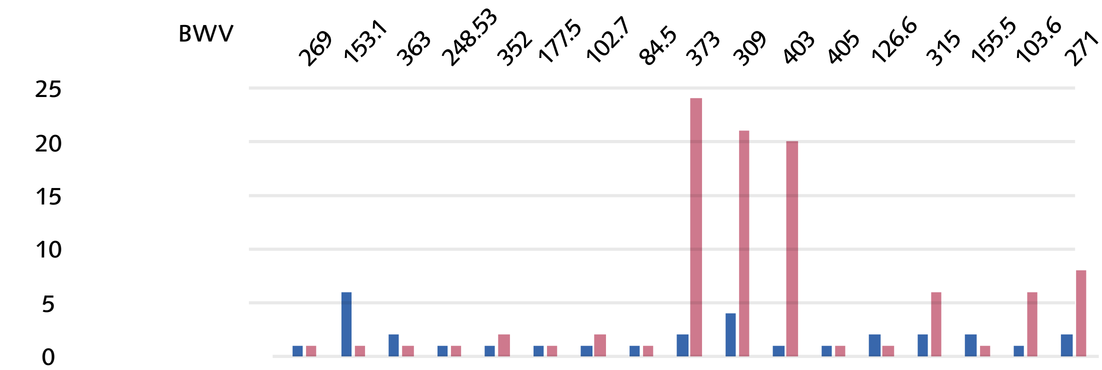



**Figure 5.2:** Example of beam search for the bass part with beam width of 2 in comparison to a greedy approach. $P$ is the total probability of the branch, $p$ the conditional local probability.

## 5.4 Training Results

As outlined in Sec. 2.1, the concept of machine learning is the derivation of knowledge from raw data and their reapplication to new and unseen data. This holds conceptual challenges: Given the necessary capacity in terms of size and number of training iterations, a deep neural network starts learning explicit characteristics of the training dataset or even memorizes it as a whole instead of generalizing and developing abstract concepts. In this case, the error measure on the training set decreases but the error on previously unseen data increases - this is called *overfitting*. [7, p. 112]

To avoid the negative effects of overfitting, the number of training iterations was determined by means of the overall probability $P$ of the whole test dataset, see Sec. 5.3. The first model (hereinafter called *model (1)*) was trained before implementing the beam search algorithm. The likelihood of the entire sequence, including the bass line and the middle parts, was calculated for the test set at every training iteration. Every 10 iterations, the state of the model with its parameters was stored. At the end, the parameter set that provided the highest sum of all total probabilities was used as the final model. *Model (1)* has proved to perform best in the described sense after beeing trained for 870 epochs.




| BWV | 269 | 153.1 | 363 | 248.53 | 352 | 177.5 | 102.7 | 84.5 | 373 | 309 | 403 | 405 | 126.6 | 315 | 155.5 | 103.6 | 271 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Width for bass part | 1 | 6 | 2 | 1 | 1 | 1 | 1 | 1 | 2 | 4 | 1 | 1 | 2 | 2 | 2 | 1 | 2 |
| Width for middle parts | 1 | 1 | 1 | 1 | 2 | 1 | 2 | 1 | 24 | 21 | 20 | 1 | 1 | 6 | 1 | 6 | 8 |

**Figure 5.3:** This chart shows the beam widths for *model (2)* for bass part and middle parts that lead to the highest overall probabilites of the chorales of the validation data set.

*Model (2)* was developed later, after implementing the beam search algorithm and further improving the hyperpameters. To find the optimal beam width for each piece, the model was chosen with the following method:

1) 31 bass notes were available as a starting note, fitting in the vocal range of a bass singer. So first, the beam widths $b_{s1} \in \{1, 2, \dots, 31\}$ were successively applied to generate 31 bass line candidates. The bass line with the highest overall probability $P$ was then chosen.

2) This bass line was used to generate 24 possible middle parts with beam widths $b_{s2} \in \{1, 2, \dots, 24\}$ starting with the alto part. The best candidate was then selected as final result.

For *model (2)*, the model trained for 580 epochs provided the highest overall probability and was thus selected as final model.

It can be observed that in most cases the beam widths $b_{s1}$ and $b_{s2}$ that lead to the best results were a beam width of 1 or a beam width of 2, see Fig. 5.3. In only 7 of 17 pieces, the beam width with highest probability was equal to or higher then 6, so in many cases, a greedy algorithm (beam width 1) had the best overall likelihood. This shows that the approach of dynamically choosing an individual beam width for bass part and middle parts per piece leads to results with a higher overall probability then an approach using a fixed beam width.

# 6

# Generation Results

After training, the resulting model was used to generate chorales from original Bach soprano parts and other melodies to examine the capabilites of the model. Audio and score files of the generated chorales and other generation examples can be reviewed online at the project homepage: *http://www.cemfi.de/research/bachnet*.

## 6.1 Original Soprano Part

The soprano part of the piece BWV 103.6, "Was mein Gott will, das gscheh allzeit", was used for the first example, see Fig. 6.1. It was generated by *model (2)*, see Appendix B for a chorale generated by *model (1)*. At first glance, the resulting chorale exhibits similarities to original Bach chorales. Lydia Steiger, a music theory teacher at the Detmold University of Music, provided the following remarks after analyzing chorales generated by both models:

- In several places, voice leading rules were violated.
- The algorithm lacked sensitivity for musical tension and therefore sometimes choses a plain solution in places were a more sophisticated composition would have been more appropriate.
- The network uses common musical phrases used by J.S. Bach. In some places the algorithm splits these phrases arbitrarily across the voices.


**Figure 6.1:** Generated harmonization given the melody from "Was mein Gott will, das g'scheh allzeit" (BWV 103.6).

The implications of this analysis are discussed in Ch. 8.

## 6.2 Broader View and Limitations

### Video Game Melody

To explore the limitations of the model, other melodies were used. As an instance, the piece "Gourmet Race" was harmonized, a popular tune from the 1996 video game "Kirby Super Star". Only a fermata at the end was added to denote the ending of the piece.


**Figure 6.2:** *Gourmet Race* from *Kirby Super Star* – Jun Ishikawa, 1996
Generated chorale.

Progressing mainly in quarter notes and featuring passing tones (see the soprano part in bar 3), the melody exhibits some characteristics of late Baroque stylistic conventions, see [4, p. 228]. The generated chorale sounds subjectively pleasing, see Fig. 6.2. To even improve the result, following adjustments to the original melody have been applied to adapt the melody to the style of Bach's soprano parts [4, p. 225]:

- More fermatas were added every two bars because they were perceived as breakpoints (bar 2, 4 and 6).
- Chromatic modification were made to obtain guiding tones (F♯ instead of F♮ in bar 4 and 7).

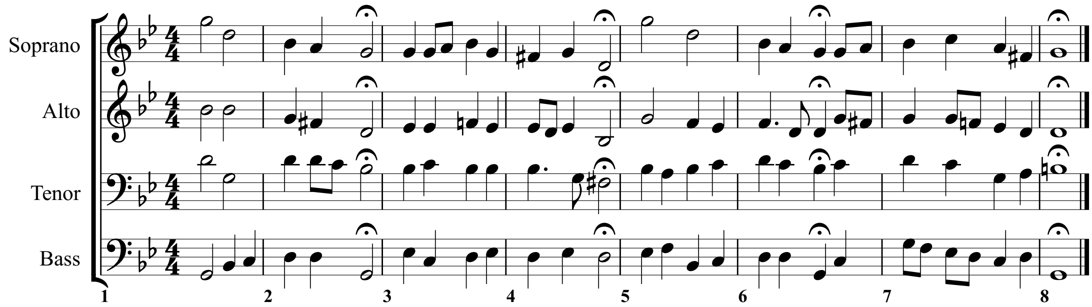



**Figure 6.3:** *Gourmet Race* from *Kirby Super Star* – Jun Ishikawa, 1996
Generated chorale with guiding tones and fermatas that were added in advance.

The result sounds more convincing, see Fig. 6.3, since, in contrast to the above example, the following can now be observed:

- In bar 2, at the added fermata, an authentic cadence $i^6_4 - V - V^7 - i$ can be seen.
- In the last bar, after the chromatic modification, a picardy third $V - I$ occurs.
- The deceptive cadence in bar 6 becomes an authentic cadence at the added fermata. It can be speculated that the network has learned to harmonize phrase ends without fermatas with a deceptive cadenc in order to maintain the tension - to what extent this is applicable to Bach's compositional style cannot be resolved at this point.

## Japanese Folk Song

The third piece discussed in this chapter is a traditional japanese folk song called "Sakura, Sakura" (Fig. 6.4). Although it has a key signature with no sharps or flats, as if it were in the key of A minor, it does not fit into late Baroque style (see Sec. 4.2). The melody was transposed up a fourth to better fit the typical tonal range of a soprano part and later transposed down again, along with the three other generated parts. Fermatas were added at the end of every second bar at the half note.

The following observations can be made:

- The network finds ways to handle the unknown structure of the melody in most places. In the final bar before the last chord, passing tones in the bass occur and lead to a dominant chord that is resolved correctly to the final tonic chord.
- Occasionally, the network does not find a valid solution. In bar 8, the phrase ends on the fermata on a sixth chord after exhibiting dissonant chords early in bar 8 - in bar 6, the phrase even ends on a dissonant major seventh chord, which is uncommon in Bach chorales [4, p. 161].

These results suggest that realistic bach-like chorales can only be obtained if the soprano part follows closely late Baroque stylistic conventions. Best results require original Bach soprano parts.


**Figure 6.4:** Sakura, Sakura (traditional japanese folk song)
Generated by the neural network.

# 7

# Online Evaluation

Evaluating computer generated music in an objective manner is an intricate task: Although voice leading mistakes were found in the musical analysis of the generation results, an artistic-aesthetic evaluation is still left to humans since the perceived degree of similarity to J.S. Bach's chorales is subjective.

## 7.1 Design

To obtain information about the perceived similarity to original chorales, an A/B test was performed via a browser based Web Audio Evaluation Tool [14]. Due to the size of the test dataset, 17 generated examples were available.

For each generated example, a pair consisting of

- the artificially generated chorale and
- the original counterpart by Bach based on the same soprano part

was prepared with a music notation program. The repeat marks were removed from the original works to adapt the structure of the generated pieces. Both the generated example and the original composition were then exported as audio files, sheet music was not provided. A playback tempo of 90 beats per minute and a virtual organ as synthesizer timbre were chosen in order to archieve a more natural sound.

For each listener, only 5 pairs were randomly chosen to maintain the listener's attention. Participants first had to give a self-assessment about their familiarity with

Bach chorale harmonization. The participants were then asked to identify the original Bach composition for each of the 5 pairs in direct comparison. Each question could be skipped. Response time and number of playbacks was not limited, modification of previously answered questions was not possible.

The network was evaluated in two stages.

1) First, a test was conducted **without beam search** using *model (1)* and addressing music majors of the Detmold University of Music. Since Bach chorale harmonization is covered in all musical fields of study, a high degree of familiarity with Bach chorale harmonization could be presupposed.

2) After improving the network and implementing **beam search**, *model (2)* was developed. Addressing professional musicologists, members of the musicology specialist group "Fachgruppe Freie Forschungsinstitute in der Gesellschaft für Musikforschung e.V.", not only could expert knowledge concerning Bach and harmonization questions be presupposed, but also a high familiarity with the *exact* Bach chorales that the network was trying to emulate.

Both groups were contacted via e-mail.

## 7.2 Results

In total, 834 answers were given, 283 in the first, 551 in the second stage. 68 music majors and 127 musicologists participated.

Of those 283 music major answers, in 61% the music majors could correctly identify the original Bach chorale. 39% misjudged the artificially generated pieces to be composed by Bach or skipped questions (see Fig. 7.1). Even though vast experience and expertise in Bach chorale harmonizations can be expected, still in only 66% of the musicologist's answers the real Bach piece was correctly identified in direct comparison. In 34% the artificially generated harmonizations was chosen or the participant gave no answer.

Looking at the evaluation results per knowledge level as seen in Fig. 7.2, more detailed insights can be gained. The best performing music major group was the group

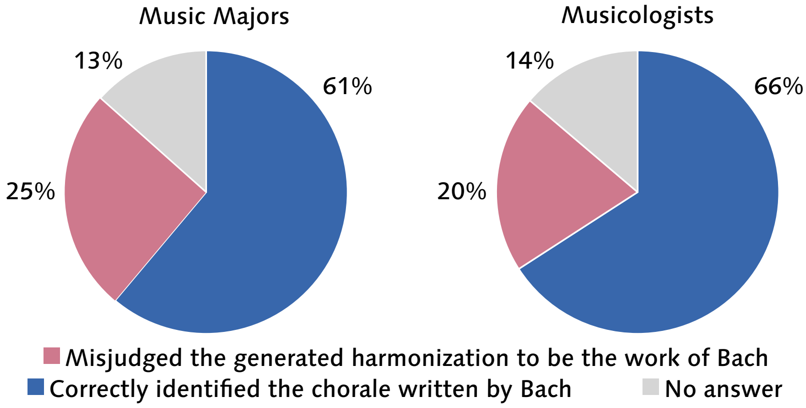



**Figure 7.1:** Summarized results of the online evaluation. The chart shows how both participating groups scored in identifying the original Bach chorale given a generated harmonization as well as the master's work.

with the highest self-assesment, giving 77% correct answers. Interestingly, concerning the music majors as well as the musicologists, the group with the lowest self-assessment performed above average: In the case of the musicologists, the group was the best performing, giving 100% correct answers, although it should be noted that since the participants were experts, the lowest self-assesment was seldom selected. In both stages, the knowledge level groups who gave the least correct answers were the groups with the middle self-assesment of 3. One can speculate that this is because it was also the default position in the web interface.

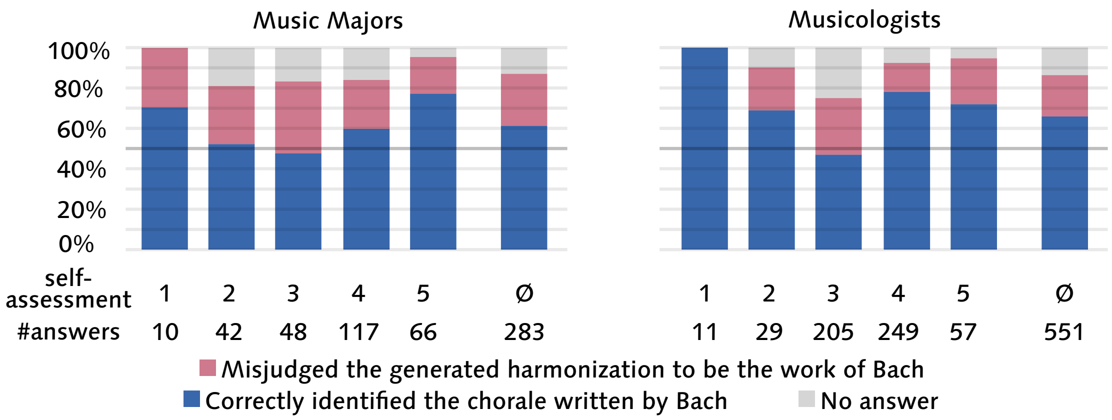



**Figure 7.2:** Results broken down by the self-assesment given. 5 corresponds to a high familiarity with Bach chorale harmonization, 1 corresponds to a low familarity.

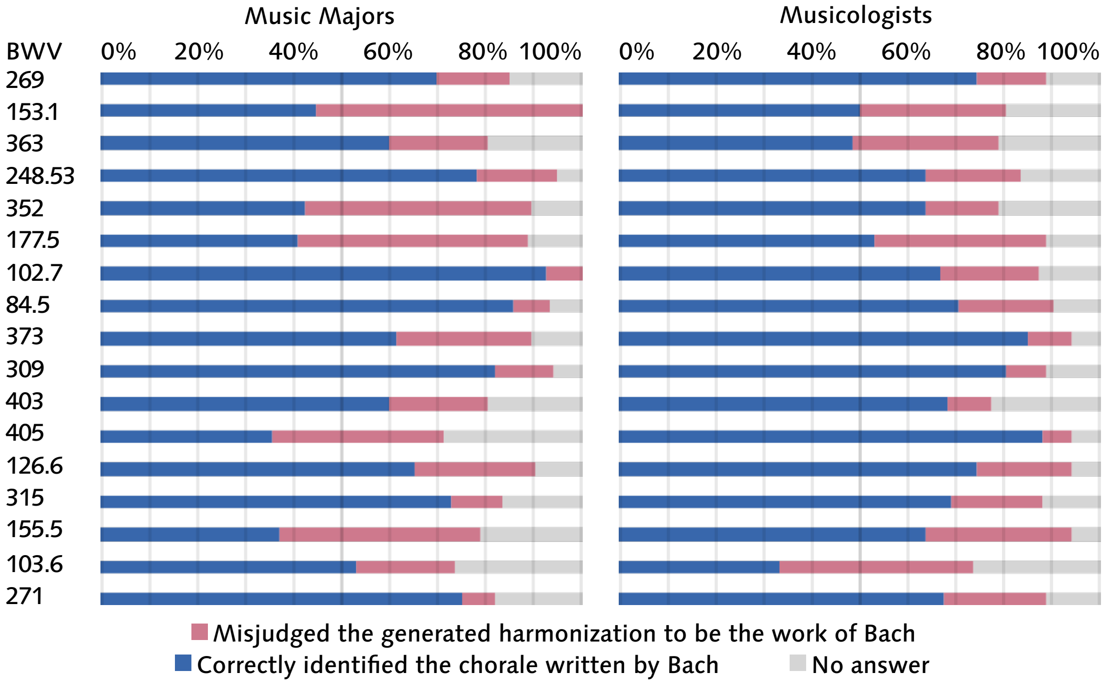



**Figure 7.3:** Results of the online evaluation per piece.

Examining the numbers per piece, one can see that some chorales were considered more often the original than others. Pieces exceeding the 50% incorrect threshold are of particular importance in this scenario: For those pieces, a participant cannot correctly recognize which is the original chorale and which was generated (random guessing).

5 of 17 generated chorales (BWV 153.1, 352, 177.5, 405 and 155.5) could not be correctly identified by the majority of the music students, as shown in Fig. 7.3. Interestingly, also 3 of 17 generated chorales (BWV 153.1, 363 and 103.6) could not be correctly identified by more than 50% of the musicology experts. The chorale BWV 103.6 was even preferred over the authentic work, with 40% of the professional musicologists considering the generated chorale the original and only 33% answering correctly - which is even better than the aforementioned random guessing ratio.

Summarizing the findings, it can be said that in many cases a bach-like sound of the generation results could be achieved. Several chorales can't be correctly identified even in direct comparison to the original work.

# 8

# Discussion and Future Work

## Summary

The performed online evaluation shows that in many cases the generated chorales sound bach-like even to experts. Thanks to a musicologically informed neural network architecture, the network is able to extract patterns and characteristics from original Bach chorales and to reapply them to new soprano melodies. The algorithm was able to derive composition concepts from the original compositions to create new harmonizations of which several were even considered more "authentic" than the actual works.

The results suggest that it's advisable to take the human creation process into account when dealing with automatic composition and harmonization based on neural networks.

## Suggestions for Improvement

Although some generated pieces are even prefered over the original work, they are not flawless in terms of voice leading. To further improve the outlined methods, musicological advice could be followed even more rigorously. A possible implementation could incorporate the anticipation of cadences: As suggested by Daniel [4, p. 159], ends of phrases could be prepared in advance as a preconnected step. Those extensions should aim at improving the obedience to voice leading rules and also the ability of the network to understand musical tension and follow accordingly. Since high quality results are only feasible if the input format matches Bach's compositional style, overcoming this restriction to only Bach chorales could also be the focus of future research. Pieces of different styles could be used for training - in order to make a more practical application for musicians possible.

Future work could also focus on understanding how the network works internally. As also suggested in [20], the question how the network develops abstractable musical skills could be examined more thoroughly. As outlined in Ch. 6, insights into the capabilities of the neural network can be used to gain understanding of Bach's compositional style.

Furthermore, the question why several generated chorales are preferred over the original composition is of interest concerning the reception history of Bach chorales that is heavily influenced by the use of his works for study purposes [11, p. 78].

# A

# Appendix Online Evaluation

In this appendix chapter, additional information about the online evaluation is presented, namely the design of the web interface and the full results.

## A.1 Web Interface

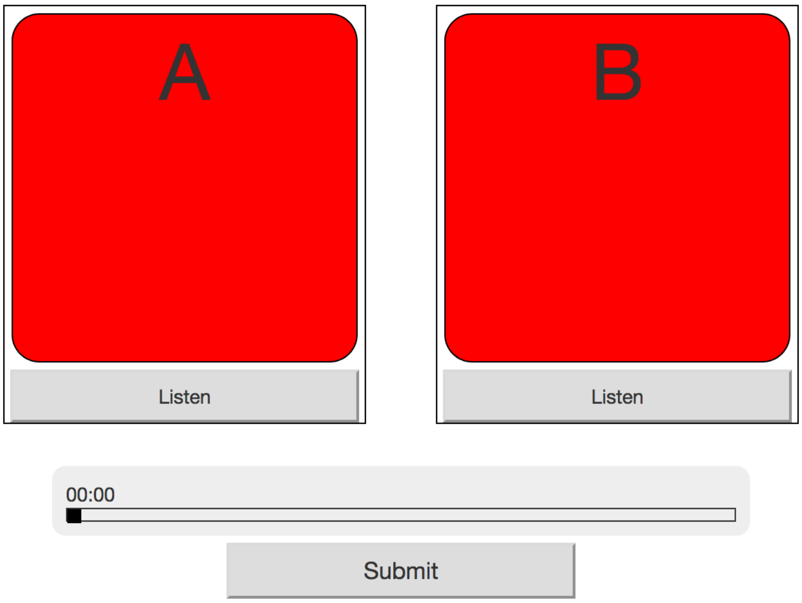



**Figure A.1:** An image of the web interface. The two alternatives were randomly assigned the buttons A and B.

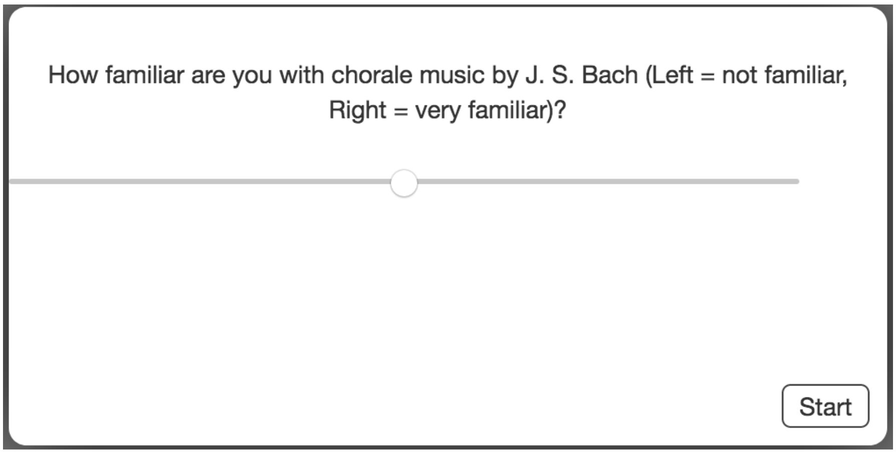



**Figure A.2:** An image of the self-assesment dialog box.

## A.2 Full Evaluation Results

### Music Majors

| BWV | Correct | Misjudged | Skipped | Not Correct | Correct | Misjudged | Skipped |
|---|---|---|---|---|---|---|---|
| 269 | 9 | 2 | 2 | 31% | 69% | 15% | 15% |
| 153.1 | 8 | 10 | 0 | 56% | 44% | 56% | 0% |
| 363 | 12 | 4 | 4 | 40% | 60% | 20% | 20% |
| 248.53 | 14 | 3 | 1 | 22% | 78% | 17% | 6% |
| 352 | 8 | 9 | 2 | 58% | 42% | 47% | 11% |
| 177.5 | 7 | 8 | 2 | 59% | 41% | 47% | 12% |
| 102.7 | 12 | 1 | 0 | 8% | 92% | 8% | 0% |
| 84.5 | 12 | 1 | 1 | 14% | 86% | 7% | 7% |
| 373 | 11 | 5 | 2 | 39% | 61% | 28% | 11% |
| 309 | 13 | 2 | 1 | 19% | 81% | 13% | 6% |
| 403 | 9 | 3 | 3 | 40% | 60% | 20% | 20% |
| 405 | 5 | 5 | 4 | 64% | 36% | 36% | 29% |
| 126.6 | 13 | 5 | 2 | 35% | 65% | 25% | 10% |
| 315 | 13 | 2 | 3 | 28% | 72% | 11% | 17% |
| 155.5 | 7 | 8 | 4 | 63% | 37% | 42% | 21% |
| 103.6 | 8 | 3 | 4 | 47% | 53% | 20% | 27% |
| 271 | 12 | 1 | 3 | 25% | 75% | 6% | 19% |
| Total | 173 | 72 | 38 | | | | |
| Avg. | 61% | 25% | 13% | | | | |

**Table A.1:** The results of the online evaluation per piece for the music students.

| Self-Assesment | Correct | Misjudged | Skipped | Total | Correct | Misjudged | Skipped |
|---|---|---|---|---|---|---|---|
| 1 | 7 | 3 | 0 | 10 | 70% | 30% | 0% |
| 2 | 22 | 12 | 8 | 42 | 52% | 29% | 19% |
| 3 | 23 | 17 | 8 | 48 | 48% | 35% | 17% |
| 4 | 70 | 28 | 19 | 117 | 60% | 24% | 16% |
| 5 | 51 | 12 | 3 | 66 | 77% | 18% | 5% |
| Total/Avg. | 173 | 72 | 38 | 283 | 61% | 25% | 13% |

**Table A.2:** The results per self-assesment level.

## Musicologists

| BWV | Correct | Misjudged | Skipped | Not Correct | Correct | Misjudged | Skipped |
|---|---|---|---|---|---|---|---|
| 269 | 26 | 5 | 4 | 26% | 74% | 14% | 11% |
| 153.1 | 15 | 9 | 6 | 50% | 50% | 30% | 20% |
| 363 | 16 | 10 | 7 | 52% | 48% | 30% | 21% |
| 248.53 | 23 | 7 | 6 | 36% | 64% | 19% | 17% |
| 352 | 21 | 5 | 7 | 36% | 64% | 15% | 21% |
| 177.5 | 18 | 12 | 4 | 47% | 53% | 35% | 12% |
| 102.7 | 20 | 6 | 4 | 33% | 67% | 20% | 13% |
| 84.5 | 21 | 6 | 3 | 30% | 70% | 20% | 10% |
| 373 | 27 | 3 | 2 | 16% | 84% | 9% | 6% |
| 309 | 28 | 3 | 4 | 20% | 80% | 9% | 11% |
| 403 | 21 | 3 | 7 | 32% | 68% | 10% | 23% |
| 405 | 28 | 2 | 2 | 13% | 88% | 6% | 6% |
| 126.6 | 23 | 6 | 2 | 26% | 74% | 19% | 6% |
| 315 | 22 | 6 | 4 | 31% | 69% | 19% | 13% |
| 155.5 | 21 | 10 | 2 | 36% | 64% | 30% | 6% |
| 103.6 | 10 | 12 | 8 | 67% | 33% | 40% | 27% |
| 271 | 23 | 7 | 4 | 32% | 68% | 21% | 12% |
| Total | 363 | 112 | 76 | | | | |
| Avg. | 66% | 20% | 14% | | | | |

**Table A.3:** The results of the online evaluation per piece for the music majors.

| Self-Assesment | Correct | Misjudged | Skipped | Total | Correct | Misjudged | Skipped |
|---|---|---|---|---|---|---|---|
| 1 | 11 | 0 | 0 | 11 | 100% | 0% | 0% |
| 2 | 20 | 6 | 3 | 29 | 69% | 21% | 10% |
| 3 | 97 | 57 | 51 | 205 | 47% | 28% | 25% |
| 4 | 194 | 36 | 19 | 249 | 78% | 14% | 8% |
| 5 | 41 | 13 | 3 | 57 | 72% | 23% | 5% |
| Total/Avg. | 363 | 112 | 76 | 551 | 66% | 20% | 14% |

**Table A.4:** The results per self-assesment level.

# B

# Appendix Analysed Chorales

In this appendix chapter, the second chorale that was analysed by music theory teacher Lydia Steiger is showed to provide an additional ouput example generated by *model(1)*.

# Ich ruf zu dir, Herr Jesu Christ (71)

Melody: J.S. Bach
Arrangement: BachNet (2019)

**Figure B.1:** The piece BWV 177.5 as generated by the neural network presented in the way it was given to Lydia Steiger for the analysis. The number 71 refers to the internal numbering system.

# Bibliography


[1] M. Allan and C. Williams. *Harmonising Chorales by Probabilistic Inference.* In: *Advances in Neural Information Processing Systems* 17 (2005).

[2] N. Boulanger-Lewandowski, Y. Bengio, and P. Vincent. *Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription.* In: *arXiv:1206.6392 [cs, stat]* (June 2012).

[3] M. S. Cuthbert and C. Ariza. *music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data.* In: *11th International Society for Music Information Retrieval Conference (ISMIR 2010)* (2010).

[4] T. Daniel. *Der Choralsatz bei Bach und seinen Zeitgenossen: Eine historische Satzlehre.* Köln-Rheinkassel, Germany: Verlag Dohr, 2000.

[5] K. Ebcioglu. "An Expert System for Chorale Harmonization". In: *Proceedings of the 5th AAAI National Conference on Artificial Intelligence.* Philadelphia, Pennsylvania, 1986.

[6] D. Eck and J. Schmidhuber. *A First Look at Music Composition using LSTM Recurrent Neural Networks.* In: *Instituto Dalle Molle di studi sull' intelligenza artificiale* (2002).

[7] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning.* The MIT Press, 2016.

[8] G. Hadjeres, F. Pachet, and F. Nielsen. "Deepbach: A Steerable Model for Bach Chorales Generation". In: *Proceedings of the 34th International Conference on Machine Learning.* 2017.

[9] H. Hild, J. Feulner, and W. Menzel. "HARMONET: A Neural Net for Harmonizing Chorales in the Style of J. S. Bach". In: *Advances in Neural Information Processing Systems 4.* Morgan-Kaufmann, 1992.

[10] S. Hochreiter and J. Schmidhuber. *Long Short-Term Memory.* In: *Neural computation* 9 (1997).

[11] Hochschule für Musik und Theater Rostock et al., eds. *Johann Sebastian Bach und der Choralsatz des 17. und 18. Jahrhunderts.* German. 2013.

[12] C.-Z. A. Huang et al. "Counterpoint by Convolution". In: *Proceedings of ISMIR 2017.* 2017.

[13] C.-Z. A. Huang et al. *The Bach Doodle: Approachable music composition with machine learning at scale.* In: *ArXiv* abs/1907.06637 (2019).

[14] N. Jillings et al. "Web Audio Evaluation Tool: A browser-based listening test environment". In: *12th Sound and Music Computing Conference.* June 2015.

[15] U. Kaiser. *Der vierstimmige Satz: Kantionalsatz und Choralsatz.* 4th. Kassel, Germany: Bärenreiter, 2015.

[16] D. P. Kingma and J. Ba. *Adam: A Method for Stochastic Optimization.* In: *CoRR* abs/1412.6980 (2014).

[17] G. Klambauer et al. *Self-Normalizing Neural Networks.* In: *Advances in Neural Information Processing Systems* 30 (2017).

[18] S. Lattner, M. Grachten, and G. Widmer. *Imposing higher-level Structure in Polyphonic Music Generation using Convolutional Restricted Boltzmann Machines and Constraints.* In: *Journal of Creative Music Systems* 2.1 (Mar. 2018).

[19] A. Leemhuis, S. Waloschek, and A. Hadjakos. "Bacher than Bach? On Musicologically Informed AI-based Bach Chorale Harmonization". In: *12th International Workshop on Machine Learning and Music.* Würzburg, Germany, 2019.

[20] F. T. Liang et al. "Automatic Stylistic Composition of Bach Chorales with Deep LSTM." In: *Proceedings of the 18th International Society for Music Information Retrieval Conference.* Suzhou, China, 2017.

[21] M. C. Mozer and T. Soukup. *Connectionist Music Composition Based on Melodic and Stylistic Constraints.* In: *Advances in Neural Information Processing Systems* 3 (1991).

[22] P. Norvig. *Paradigms of Artificial Intelligence Programming: Case Studies in Common Lisp.* Elsevier Science, 1992.

[23] A. Paszke et al. *Automatic differentiation in PyTorch*. In: *NIPS Workshop Autodiff* (2017).

[24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Learning representations by back-propagating errors*. In: *Nature* 323.6088 (Oct. 1986).

[25] S. Suzuki and T. Kitahara. *Four-part Harmonization Using Bayesian Networks: Pros and Cons of Introducing Chord Nodes*. In: *Journal of New Music Research* 43.3 (2014).

[26] G. P. Telemann. *Fast allgemeines Evangelisch-Musicalisches Lieder-Buch*. Hamburg, Germany: Philip Ludwig Stromer, 1730.

[27] P. M. Todd. *A Connectionist Approach to Algorithmic Composition*. en. In: *Computer Music Journal* 13.4 (1989).

[28] R. P. Whorley et al. *Harmonising Melodies: Why Do We Add the Bass Line First?* In: *Journal of New Music Research* (2013).