Comparing a Q-Learning agent's and human-generated piano fingerings

Bachelor Thesis – 7,5 ECTS
02/07/2021

Under supervision of:
Francisca Pessanha &
David Terburg

Tim Koornstra
Student number: 6435777

Universiteit Utrecht

Bachelor Artificial Intelligence
Faculty of Humanities
Utrecht University
Netherlands

Table of contents

1. Abstract

This paper reports the findings of a Q-Learning reinforcement learning agent when given the task to generate the fingering for piano sheet music. Although some studies on automatically generating piano fingering have been done, these researches had not focused on reinforcement learning and Q-Learning, specifically. An environment, a set of states and actions, and a reward scheme had been created for the Q-Learning agent, and it had been given four piano sheets, which were used to generate fingerings. The input for the algorithm contained only right-handed, single-note melodies. The results showed that the algorithm had an overall better performance than previous research done on this topic, with some limitations. These results lend support to the idea that the agent learns not only to optimally place its fingers but also some hand-ergonomic rules, which it was not taught. The research demonstrates that reinforcement learning is a tool that can be used for newly-beginning pianists to help them with an understanding of piano fingerings.

2. Introduction

A study by the Royal Philharmonic Orchestra in the United Kingdom has shown that 9 in 10 British children either played or wanted to play a musical instrument (Royal Philharmonic Orchestra, 2019). However, only 7 in every 10 children say they can indeed play a musical instrument (Royal Academy of Music, 2014). There have always been a lot of people who wish to learn a new instrument, but do not have the time, the means, or discipline to take lessons. With the rise of the internet, new ways to learn an instrument have emerged. There are countless apps in the app store that can teach one how to play the guitar, and there are many more tutorials to be found on video services, such as YouTube. Because of this novelty, some aspects are suboptimal or even completely missing, and are not on the level where they can "support the strict requirements of a professional music education context" (Eremenko, Morsi, Narang, & Serra, 2020). When learning to play the piano, for example, one of the most important things to have is a good finger allocation. Even Frédéric Chopin – one of the most famous classical pianists and composers – said "Everything is a matter of knowing good fingering. […] One needs only to study a certain position of the hand in relation to the keys to obtain with ease the most beautiful quality of sound, to know how to play short notes and long notes, and [to attain] unlimited dexterity" (Eigeldinger, 1988). Although there are good videos and websites out there that explain the basics, one can only learn this by practicing. This research is about generating piano fingerings from sheet music so that new players can have a reference, even when a teacher is not around to help them.

2.1. Reinforcement Learning

Learning can be split into three paradigms: supervised learning, unsupervised learning, and reinforcement learning. There have been multiple successful studies that tried to generate finger allocations from a music sheet, using machine learning techniques as Hidden Markov Models (HMM) and deep learning (Nakamura, Saito, & Yoshii, 2020), and dynamic programming (Andersson & Håkansson, 2014). These are all examples of supervised learning. This research will use a different approach and will use reinforcement learning (RL). The idea behind reinforcement learning is that there is an agent that can perform certain actions with its environment. By trying different actions, the agent will get a reward based on a certain state it is in. Within reinforcement learning, there are multiple learning algorithms, all of which are model-free (this means that it does not use a transition probability distribution associated with Markov Decision Processes). This paper will use the Q-learning algorithm. This algorithm trains the agent based on the value of an action in a given state. This means that there is a reward scheme for the agent, and a mathematic equation is used to calculate the Q-value of a state, given the action and reward scheme.
There are multiple reasons for choosing reinforcement learning for this study. Firstly, unlike supervised learning, machine learning does not require the output to already be known. This is an interesting concept since this means that the agent has to start from scratch, which can be likened to a student that just started. It also means that we can track the performance of the agent between training sessions to see if it improves without supervision. Just like with a human without guidance, the agent will try to learn the most optimal way of playing notes, such that a music piece can also be played in faster tempos without the risk of getting its "fingers" all knotted up and losing accuracy. Another reason why reinforcement

learning is an interesting approach for this study is that – as mentioned above – no other research has used this method yet. This study will shed some light on the concept from a different perspective of computer science and artificial intelligence.

2.2. Research Question

This paper researches the effectiveness of reinforcement learning when learning piano fingering from sheet music. Not only can this research assist other research concerning this topic, but it can also be used to help new students to learn to play the piano. To achieve this goal, the following main research question is relevant: **"How does a reinforcement learning agent compare to a human in generating optimal piano fingerings?"**. To be able to answer this question in the most well-defined way possible, this research paper will explain the algorithm used to train and test the reinforcement learning agent, as well as how it is set up and used. Some goals of this paper include creating an agent that can generate piano fingering well and can be used by students, and seeing if this reinforcement learning agent learns while retaining high accuracy.

2.3. Outline

The next section in this paper will describe some related work concerning piano fingering and reinforcement learning to give a good idea of the technicalities. The following section (section 3) will describe the methods used in this paper. This will include a full explanation of the algorithm design, as well as a rundown of the data used, information about how training and testing were performed, as well as a description of the evaluation of the model. Section 4 reports the results of the research, after which the next section will discuss these results in depth. The discussion also includes the resemblance and dissimilarities between the results of this paper and the results of previous research, together with the impact this research could have on future studies. The last section contains a quick recap of the study and some conclusions that can be drawn from it.

3. Related work

3.1. Piano Fingering

Since the invention of the pianoforte by Bartolomeo Cristofori di Francesco in 1700, the techniques for piano fingering have not changed much, and thus there are very few (scientific) publications on this topic. The only thing that has changed since the invention is the notation, which changed at the beginning of the 20th century (English Fingering, 2001). Fingering for a musical instrument, in general, has been described as "(1) A system of symbols (usually Arabic numbers) for the fingers of the hand (or some subset of them) used to associate specific notes with specific fingers.... (2) Control of finger movements and position to achieve physiological efficiency, acoustical accuracy [frequency and amplitude] (or effect) and musical articulation" (Rendel, 2003). The first element, the notation, is straightforward: the thumb is the first finger and is thus denoted as a "1", the index finger is the second finger, and is thus written as "2", et cetera. The difficult part is the second element: to learn the fingering itself. Someone who has never played the piano before but can read notes and find them on the piano would have great trouble finding an optimal way of playing the most basic, white key-only C major scale as shown in figure 1. This is because it involves passing the thumb under the other fingers, such that an optimal new starting location is found for the next set of notes. It is only after some practice with this particular music part that they find an effective fingering. And even so, they would not be able to translate this knowledge in finding a fingering for Chopin's *Nocturne Op.9 No.2*, for example. To learn this skill, one needs to have had a lot of practice.

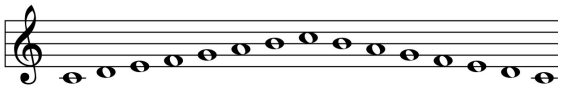

Figure 1 - C Major Scale (Up and Down)

3.2. Machine Learning

To simulate this learning process we can use machine learning. As mentioned before, there are three machine learning paradigms: supervised learning, unsupervised learning, and reinforcement learning. The task of supervised learning is to create a function that maps an input to an output based on labeled input-output pairs (Russell & Norvig, 2010). That is, we have ground truth annotations for the data in the training set. In unsupervised learning, data is not tagged by a human, but it captures patterns as neuronal predilections or probability densities (G. & Sejnowski, 1999). Lastly, there is reinforcement learning. This does not need labeled input-output pairs – as with supervised learning – and does not need sub-optimal actions to be corrected. It instead searches for a balance between exploration and exploitation (Kaelbing, Littman, & Moore, 1996). A key difference between reinforcement learning and unsupervised learning is that reinforcement learning is reward-based, whereas unsupervised learning is based on clustering and the association of data.
To differ from and be able to compare results with the supervised machine learning methods used in other

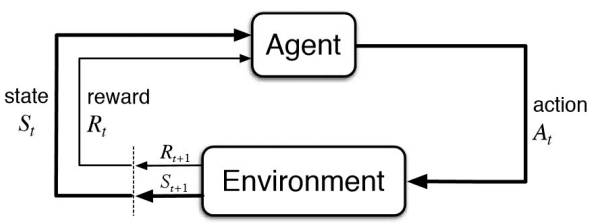

Figure 2 - Reinforcement Learning Cycle

research, such as HMM and deep learning (Nakamura, Saito, & Yoshii, 2020) and dynamic programming (Parncutt R. , Sloboda, Clarke, Raekallio, & Desain, 1997, Vol 14), this paper uses reinforcement learning. Reinforcement learning is modeled as a Markov Decision Process (MDP), meaning that there are a set of states and a set of actions for each state. The reinforcement learning agent can interact with the environment by performing an action, which results in a (numerical) reward and a new state. The specific algorithmic design used for this paper is described in the next section.

4. Method

4.1. Design

This subsection describes the design of reinforcement learning and everything else that is necessary for it to function properly. The code for this algorithm has been written in Python. Even though there are a lot of libraries out there for reinforcement learning, such as OpenAI Gym (OpenAI, 2021), this research uses a from-scratch implementation of reinforcement learning.

4.1.1. Data

The input data that the reinforcement learning agent uses is a music sheet. Using image recognition to parse notes from sheet music is too costly and too inaccurate. The site *MuseScore* (MuseScore, 2021) offers free downloadable piano sheets, and more specifically, the sheet in MusicXML file format. This file format is based on the Extensible Markup Language (XML) format. XML is a markup language that describes a set of rules for encoding documents in a format that is both human-readable and machine-readable. MusicXML uses this principle to describe musical notation. For the algorithm to be able to use this data, the data needed to be parsed. This was done using the Document Object Model (DOM), which represents the XML document as a tree-like structure. The notes were then read from the MusicXML file and the relevant information that was stored in the note was saved in a Pandas data frame. The relevant information consisted of the step, the alter, and the octave. The middle C would thus be parsed as having step C, alter none, and octave 4. Half a note above that would be parsed as step C, alter #, and octave 4. Because working with numbers rather than strings is easier when it comes to reinforcement learning, these notes were then converted to distance from the middle C, based on physical distance, rather than musical frequency. C4 would become 0, C#4 would be 1, and B3 would be -2.
Because of the limited time for this paper, a decision had to be made about which exact data would be used for the reinforcement learning agent. Because of the complexity and the focus on starting pianists, only the right-handed single melodic notes were used. That means that multi-note chords were not used as data, as this would increase the complexity of the research by a substantial amount. Data for the left hand was not included, since this would be a mirrored version of the right hand, using different notes. When playing the piano, the fingering of one hand is very rarely dependent on the fingering of the other hand because the hands do not impede each other. However, it might be interesting for future work to discover if the reinforcement learning agent would focus more attention on the right hand like it is the case for humans (Parncutt, Sloboda, & Clarke, Interdependence of Right and Left Hands in Sight-Read, Written, and Rehearsed

Fingerings of Parallel Melodic Piano Music, 2011). Due to time and increased complexity constraints, however, this is not in the scope of this paper.

4.1.2. Environment

The environment for the agent should be the same as for a pianist. That means that the agent should have a piano to play and a hand with which he can play the piano. Each note on the piano was described as a floating-point value which represents the physical distance to the middle C (C4). The hand was modeled as 5 fingers, all of which have a position on the piano (i.e. they all have a floating-point value assigned to them). The actions that the agent could perform were based on that hand. Each of the fingers represented an action. Playing a note with the thumb was described as action one, the index finger as action two, et cetera. After "playing" a note, the hand got relocated to have new positions for each finger. The hand was placed based on the previously played note, as well as the finger used to play that note. It was decided that the key to center around was the natural (the closest white key) of the note played. The finger that played this note would be relocated to the original note, whereas the other fingers were placed a certain amount of full notes away from the natural played note. The example shown in figure 3 shows that the middle finger played the note Eb5. That means that the middle finger got the new location of Eb5. The natural of Eb5 is E5. The fingers adjacent to the middle finger, the index and ring fingers, got the neighboring keys, D5 and F5, respectively, as a new position. The thumb got the key next to D5, which is C5, and the last finger got the key next to F5, which is G5.

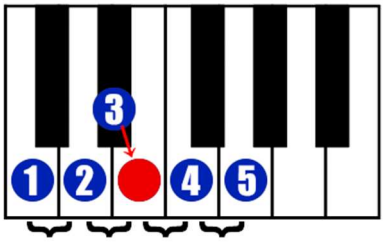

Figure 3 - Example finger relocation. The note played with the third finger gets "converted" to the nearest white key, and the other fingers are relocated according to this natural.

Previous research (Sloboda, 1974) has shown that for advanced pianists, the number of notes held in working memory in advance of the note currently being played is around six. Because this research focuses on pianists that are not at all skilled in playing the piano, the set of states contained a triple that consisted of the previous note played, the current note played, and the next note to be played. Having states that contained more notes would have resulted in a much more complex environment.

4.1.3. Algorithm

Out of the many reinforcement learning algorithms to chose from, this paper used the Q-Learning algorithm, and specifically the Epsilon-Greedy algorithm. The goal of this algorithm and its agent is to maximize its total reward. Each performed action generates a reward based on the current state. The reward is calculated using a straight-forward function: take the distance from where the finger we use previously was and where we will place the finger now, and multiply this subtraction by -1. We multiply by -1 since we want to maximize the score, and a big stretch should thus not be rewarded but penalized. This reward is then used to calculate the Q-Value based on the Bellman Equation. This equation is a simple

value iteration update, using the weighted average of the old value and the new information. This function is described as follows:

$$Q^{new}(s_t, a_t) \leftarrow Q^{old}(s_t, a_t) + \alpha \cdot \left(r_t + \gamma + maxQ(s_{t+1}, a) - Q^{old}(s_t, a_t)\right)$$

α is the learning rate ($0 < \alpha \leq 1$), r_t is the calculated reward, γ is the discount factor ($0 \leq \gamma \leq 1$) – this has the effect that the earlier values are received as higher than those received later, and $maxQ(s_{t+1}, a)$ is the estimate of the optimal future value. These values were saved in an array that contained a large range of state-action pairs (which were the note-triples paired with the used finger).

4.1.4. Training and Testing

Training in the Epsilon-Greedy algorithm happens using a combination of exploration (of a random action) and exploitation (of previous knowledge). Before training, a value for epsilon ($0 < \epsilon \leq 1$) gets determined. This value determines how much of the training happens using exploration and how much happens using exploitation. A value of $\epsilon = 0.9$ means that 90% of the training is done by taking the best possible action saved in the Q-Values table, while 10% of the training is done by taking a random action saved in the Q-Values table. This value for epsilon was also used in this paper.
After determining this value for epsilon, the agent will train using the data set for a specified amount of episodes (1000 for this research). For every note, we will determine if we use exploration or exploitation (based on epsilon). If we use exploration, we simply take a random finger as action. If we use exploitation, we take the argument with the highest Q-Value in our table. We then calculate the reward for performing this action, after which we immediately relocate our hand. This is done by placing the fingers on the adjacent notes. For example, finger 3 plays C4, finger 1 will be on A3, finger 2 will be on B3, finger 4 will be on D4, and finger 5 will be on E5. After this readjustment, we calculate our Q-Value using the aforementioned Bellman Equation and store this in the table.
Testing the data work similar to the last few steps of training: we get the best action given the state and append that action to the list of actions. Now we have the list of best actions for a music sheet using the Epsilon-Greedy Q-Learning algorithm.

4.2. Evaluation

The optimal piano fingering for a piano piece is not an objective matter. It is a combination of hand size, flexibility, and ergonomics, all of which differ from person to person and even from hand to hand. To evaluate the results of the algorithm, human-labeled piano sheet music was used. The computer-labeled output will be compared to the human-labeled output and feedback will be given. To also give a numerical score to the output, the agent will also be evaluated using an accuracy measure, which will be compared to the human-based input.
Figures 4, 5, 6, and 7 show the collection of data that was used for the evaluation of the fingering. Figures 4 and 5 were selected as test data based on their appearances in previous work (Andersson & Håkansson, 2014) (Nellåker & Lu, 2014), such that the data could easily be compared to this work. Sheet 4 is the beginning of *Eine Kleine Nachtmusik* by Wolfgang Amadeus Mozart, and sheet 5 is the beginning of *The Entertainer* by Scott Joplin.

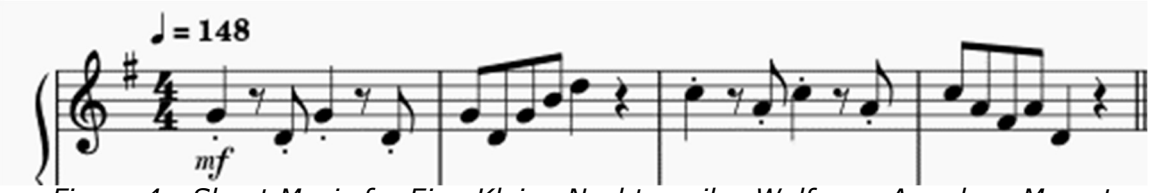

Figure 4 - Sheet Music for Eine Kleine Nachtmusik - Wolfgang Amadeus Mozart

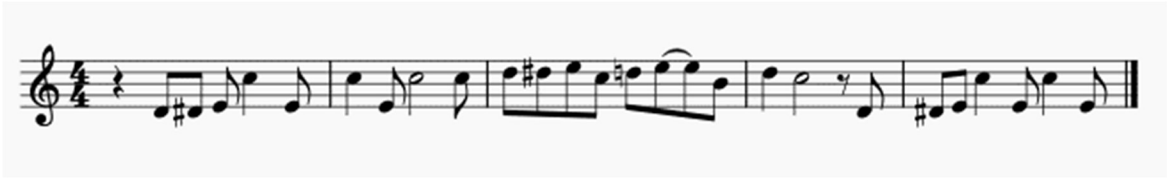

Figure 5 - Sheet music for The Entertainer - Scott Joplin

To reflect a contrast between different music styles, figures 6 and 7 were chosen based on their differences in dynamics. Sheet 6 is the start of *Für Elise* by Ludwig von Beethoven, and sheet 7 is the beginning of Chopin's *Nocturne Op.9 No.2* Sheet 6 contains notes that are physically very close to each other, whereas sheet 7 is a typical Chopin piece and includes notes that are spaced far apart, where the player's hand would need to "jump" to the next note, thus allowing for a more dynamic sound.

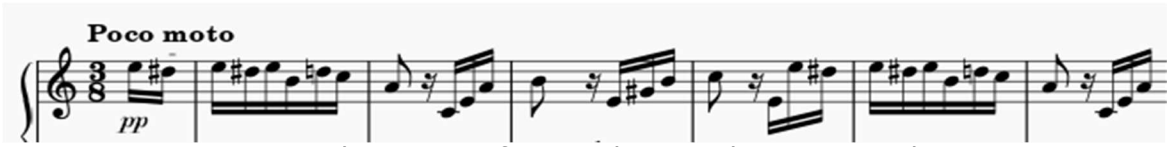

Figure 6 - Sheet music for Für Elise - Ludwig von Beethoven

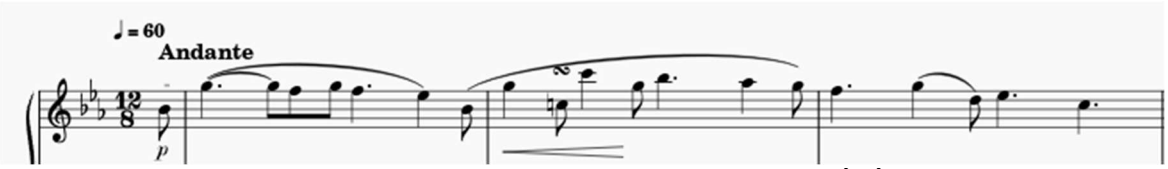

Figure 7 - Sheet music for Nocturne Op.9 No.2 - Frédéric Chopin

Because reinforcement learning is model-free (i.e. it is not able to generalize over data it has not seen before), this paper describes the results of training over only the test set, as well as training over everything but the test set. This will illustrate the power of Q-Learning on data it has already seen, while on the other hand provide examples of the limitations of reinforcement learning with data it has not seen yet. In the case that the state has not previously been seen by the agent, it will get the first best action from the Q-Value table, which, in this case, is the thumb. This is because the Q-Values table is initialized to have all values as 0. That means that every state that the agent has not seen will have a Q-Value of 0 for each finger. When all the highest values are equal, the algorithm picks the first argument with that score.

5. Results

This section will show the results of the algorithm based on the data set described in the previous section. Each subsection is dedicated to a melody. The first subsection will list the results for *Eine Kleine Nachtmusik*, the second subsection presents the results for *The Entertainer*, subsection 3 will show the results for *Für Elise*, and the last subsection will contain the results for *Nocturne Op.9 No.2*. Each of these subsections contains three different results. The first figure shows the agent-labeled melody when tested on the training set. This means that the agent had at least been trained on the data it was tested on. The second figure shows the results of the algorithm when the test data was not part of the training data. That is, the agent had not been trained on the data that it was tested on. To allow the results to be compared to the human-labeled results easily, these results are represented in the third figure. To allow the reader to see the differences more clearly, the red-colored fingering in the agent-labeled results represents fingerings that differ from the human-annotated data.

5.1. Eine Kleine Nachtmusik

This subsection shows the results for *Eine Kleine Nachtmusik*. In future references, this sheet will be referred to as "Melody 1".

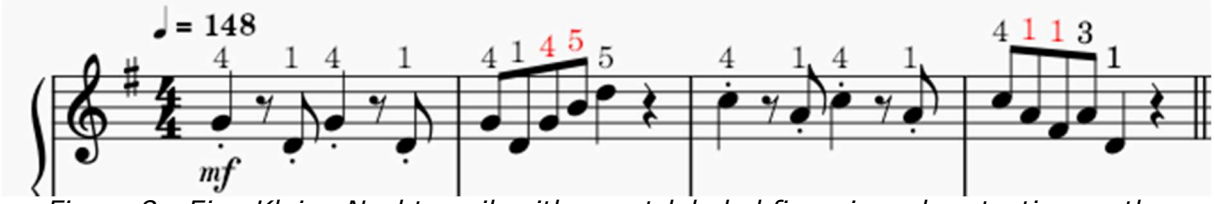

Figure 8 - Eine Kleine Nachtmusik with agent-labeled fingering when testing on the training set

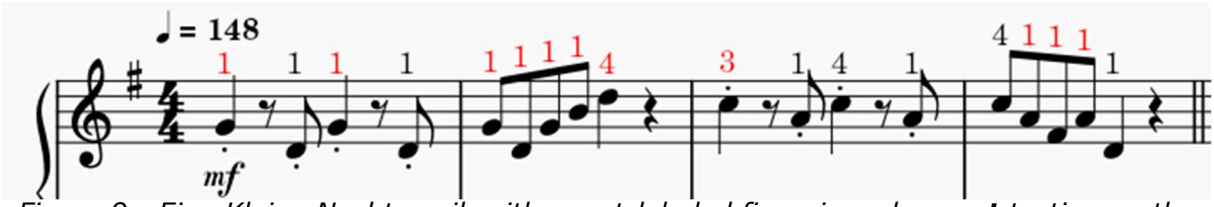

*Figure 9 - Eine Kleine Nachtmusik with agent-labeled fingering when **not** testing on the training set*

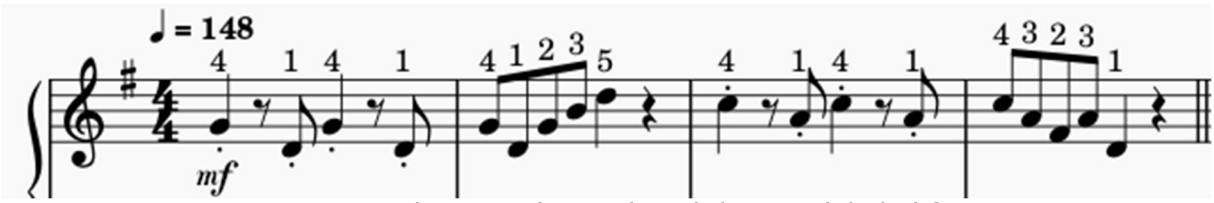

Figure 10 - Eine Kleine Nachtmusik with human-labeled fingering

5.2. The Entertainer

This subsection shows the results for *The Entertainer*. In future references, this sheet will be referred to as "Melody 2".

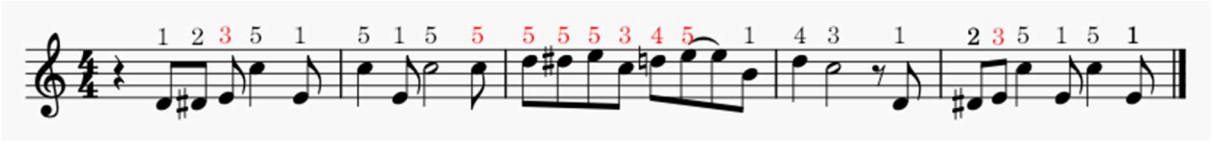

Figure 11 - The Entertainer with agent-labeled fingering when testing on the training set

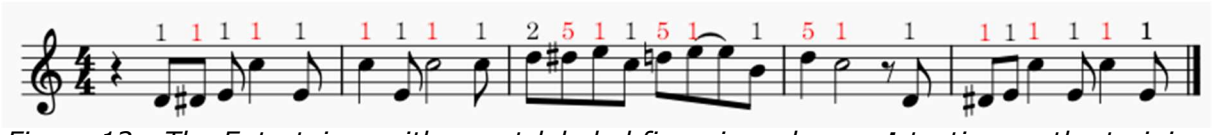

*Figure 12 - The Entertainer with agent-labeled fingering when **not** testing on the training set*

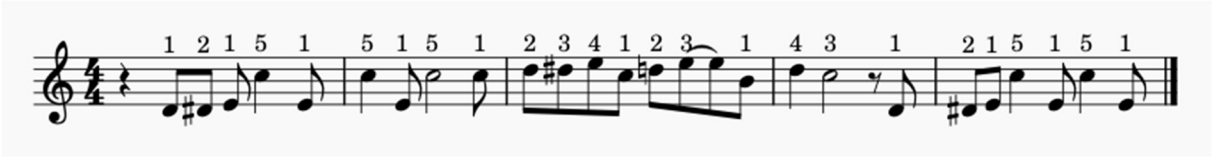

Figure 13 - The Entertainer with human-labeled fingering

5.3. Für Elise

This subsection shows the results for *Für Elise*. In future references, this sheet will be referred to as "Melody 3".

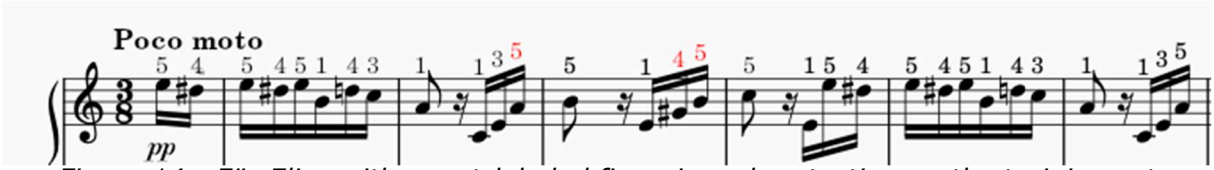

Figure 14 - Für Elise with agent-labeled fingering when testing on the training set

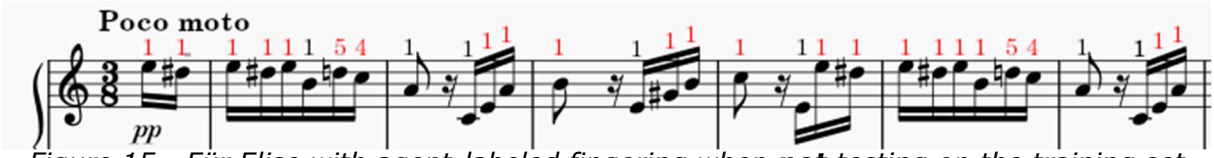

*Figure 15 - Für Elise with agent-labeled fingering when **not** testing on the training set*

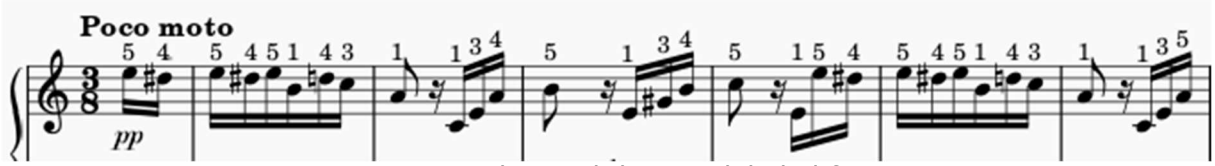

Figure 16 - Für Elise with human-labeled fingering

5.4. Nocturne Op.9 No.2

This subsection shows the results for *Nocturne Op.9 No.2*. In future references, this sheet will be referred to as "Melody 4".

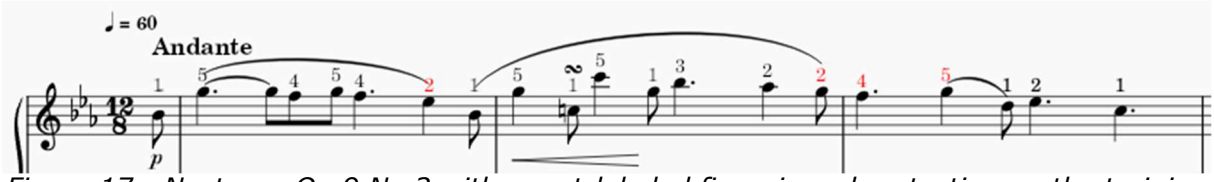

Figure 17 - Nocturne Op.9 No.2 with agent-labeled fingering when testing on the training set

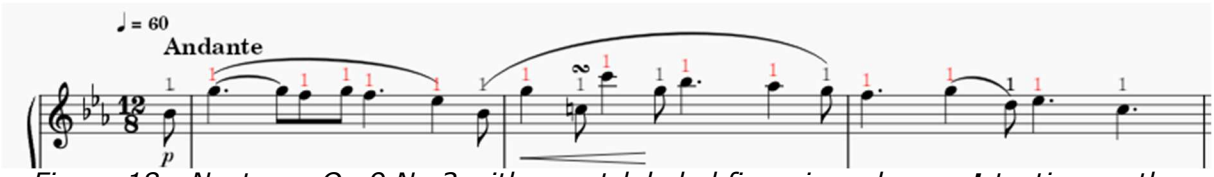

*Figure 18 - Nocturne Op.9 No.2 with agent-labeled fingering when **not** testing on the training set*

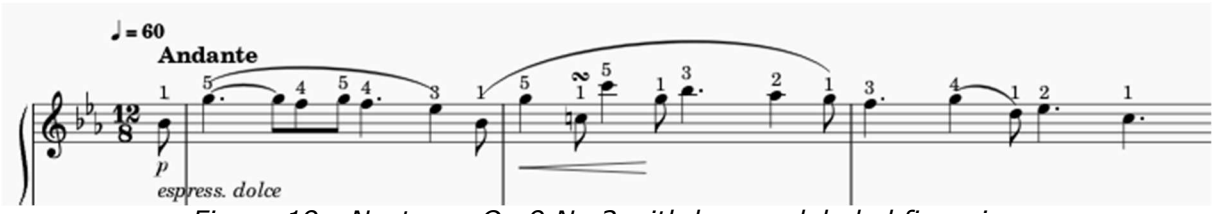

Figure 19 - Nocturne Op.9 No.2 with human-labeled fingering

6. Discussion

This section will discuss the results as listed in the previous section. Because we want to answer the research problem here as much as possible, the first subsection will discuss the findings when compared to human results and try to explain these results. Taking the discussion from the first subsection, along with the second subsection, where we discuss the results and compare them to results of previous research, into account, this will also allow us to examine and study whether we have achieved our goals for this research, and explain why or why not. The third subsection will then describe some of the limitations of this research, which will be combined with the last subsection, to describe ideas for future work.

6.1. Comparison to Human Results

6.1.1. Eine Kleine Nachtmusik

The results for melody 1 show that the agent-labeled fingering when training on the test set (as shown in figure 8) comes very close to the human-labeled fingering (figure 10), with an accuracy score of 77.78%. There are some differences to discuss, however. The first highlighted red fingering shows that the agent would play the second measure with the fingers 4-1-4-5-5, whereas the human-labeled fingering for the same measure is annotated as 4-1-2-3-5. This can be explained by the way the hand was modeled in the algorithm. As stated in the "Environment"

subsubsection in the methods section, the new location of the hand was based on the location of the note played and the finger used to play that note. When the second note in the second measure was played, the fingers had the positions: D5-E5-F5-G5-A5. We can understand why the agent played G5 with the ring finger, because that finger was already on that note, meaning that it would have a distance and score of 0 to play that note, which is always higher than using any other finger. From there, playing B5 with the last finger is also explainable, since the last finger was on A5, meaning that the finger closest to that note was indeed that finger. This would thus have been a good finger to use, given that the third note was played using the fourth finger. But because this was not the most optimal fingering, this finger had been labeled as "wrong", even though in that circumstance, it was, de facto, the right finger to play.
The second set of errors (the ones in the fourth and final measure) are of a different nature. The first error could be because the agent uses the epsilon-greedy algorithm and thus used exploration to find that finger 1 was the best note and kept iterating on that. Of course, that would have to mean that by a random chance, the algorithm never had a chance to explore the third finger, which would be a very slim chance, given the fact that every note had a 10 percent chance to be randomly explored every episode, leaving about a 1 in 100 chance that this action had not at least been tried. The error after that, however, is more easily explainable. Given the fact that the agent had previously played A5 using its thumb, the closest finger for F#5 would also be this thumb. This error could have been corrected by implementing a rule that reduced a penalty for the agent to pass another finger over its thumb when playing a note lower than the note the thumb had previously played. However, since the algorithm was designed specifically to minimize the distance the fingers had to travel, this rule was not taken into account, and given the previous erroneously fingered note, this would have been the best finger to use in this situation.

6.1.2. The Entertainer

The data for figure 11 suggests that the agent-labeled fingering when testing on the training set for melody 2 differs substantially from the human-labeled fingering. This labeling has an accuracy of 64%, which makes it a lot lower than the accuracy for melody 1. The first minor mistake (which is the same as the last mistake) can be accredited to the fact that passing the thumb under another finger had also not been implemented as a feature. The fingering is not a necessarily "wrong" one, but rather an uncomfortable one, since three fingers are placed very closely together, which also limits the range of the hand. This would mean that someone with very small hands would have to jump notes when going from E5 to C6. This, of course, would result in a sound that is not desirable, as it would involve a minor "pause" in the right-hand melody. But since the agent has been modeled with an infinitely flexible hand, this is not taken into account, nor has the musical quality of the outputted fingering been taken into account.
The second set of errors is easily explainable. After playing the third note in the second measure, the shortest distance – which is 0 – to the next note is achieved by playing the note with the same finger, which is the last finger. The agent does not believe that switching to a finger one for the next note is worth it, since that would involve a tremendous penalty, whereas playing with the same finger is free. Because the agent has been modeled to only take into account the previously played note, the current note, and the direct next note, this would not be such a big deal in the eyes of the agent, since the calculated distance would only be 0.5

for these three notes. Had the agent been able to look forward more, however, it would have noticed that it might be better to switch now, so that we can minimize the total score, rather than just this local score. Because the agent does not switch early on, the labeling will not be corrected, since the human-labeled fingering for the first wrong note is dramatically different than the agents.

6.1.3. Für Elise

The results for melody 3, figure 14, show the highest accuracy of all melodies: 90%. This can be explained since almost half of the training set is the same. That means that if one half of the training set is almost fully correct, the second half should have the same result. It can also be explained because there is a lot of repetition in the notes, meaning that the agent has not only seen the states before but has also been trained on them many times.
There are still two mistakes to be discussed, however. Both of these mistakes are already discussed in the previous subsubsections. The first mistake deals with the hand positioning. The finger that is closest to A5 is the last because this one is on G5. One thing that could be noted is that replacing this finger with the ring finger would not result in a higher score, but this would stay the same. It is purely because the agent does not take the quality of the played piece into account that it chooses finger 5 over finger 4.
The second mistake is the same as the first, but because the first note is played with the "wrong" finger, the second note has to be wrong as well. Given the circumstance that the fourth finger is already on G#5, the closest finger to play B5 with would be the fifth finger, which would be the most optimal fingering, had the first error been correct.

6.1.4. Nocturne Op.9 No.2

The results for melody 4 when testing on the training set (figure 17) exhibit the second highest accuracy of all the melodies with an accuracy score of 78.95%. There are a few differences to discuss. The first of these differences happens in the second measure. The second finger is used to play Eb6, rather than the third finger, as can be seen in figure 19. This error can be compared to the first error in melody 2. For the agent, playing finger two would have the same score as playing finger 3, since the distance to Eb6 is both 0.5. The agent thus chooses the first argument of the Q-Values for this state, which would be the second finger. This is not a "wrong" finger, necessarily, but rather an uncomfortable one. For a human, the third finger would get nudged in between the second and fourth finger, and the stretch from Eb6 to Bb5 is not only uncomfortable for most players but even impossible for some players.
The second error is harder to explain. Using the index finger for the first note is understandable since according to the hand placement, the second finger is the closest finger to that key. As can be seen in figure 19, the next set of notes would require passing the third finger over the first finger, which, as stated previously, had not been implemented as an optimal way of fingering. However, the agent did find that crossing the fourth finger over or under the second finger would result in a better score. One can understand that this would result in a cheap operation since the fifth finger will then already be in place, and the distance to the note after that will be minimal. Having a human play this, however, is not comfortable, and might even injure the fingers. The reason that the agent did decide to play

this, is because hand ergonomics had not been taken into account when creating the reward scheme.

6.1.5. Not tested on training set

As stated previously, Q-Learning is a model-free implementation of reinforcement learning. That means that the agent cannot generalize over data it has not seen before. This is reflected very clearly in the results for melody 1 when the agent was *not* tested on the training set (as shown in figure 9). The accuracy for this set was exactly half of the previous result: 38.89% (as opposed to 77.78%). Most of the errors come from the agent using the default answer for a not-seen state: the finger 1. There are some different labels as well, however. The only way this can be explained is that the agent had been trained on these particular states and used that knowledge to label these notes. This did not always result in a good fingering, however. Only two of these fingers were labeled correctly. This is since the other labels had already been labeled incorrectly, and thus the hand placement would have been wrong, resulting in a fingering that would not seem "natural" when compared to the human-labeled fingering. These results can be found throughout all the melodies. Melody 2 has an accuracy of 48%, compared to 64% when training on the test set, melody 3 has an accuracy of 23.3%, compared to 90%, and melody 4 has an accuracy of 78.95%, compared to 36.8%. Interestingly, it seems that there is a correlation between the accuracy when training on the test set, and the accuracy when not training on the test set. A higher score on the former seems to correspond with a lower score on the latter.

6.2. Comparison to Previous Research

Because melodies 1 and 2 were chosen based on their appearance in other research, the comparison between researches can be made more easily. One of these studies (Nellåker & Lu, 2014) used dynamic programming, along with an ergonomic hand model to generate fingerings for melody 1. They found an accuracy score of 65% (compared to this paper's 77.78%). Interestingly, even though an ergonomic hand model had been implemented, some mistakes were still not fixed, even though the algorithm would pass a finger over the thumb. The results differ significantly since the researchers implemented a rule that had the algorithm avoid the use of fingers 4 and 5. Because no such rule was implemented for the reinforcement learning agent, these fingers were used more often, and the agent seemed to figure this rule out by itself, indicating that some of the rules implemented to learn ergonomics are indeed learned by the agent itself without explicitly listing them.
The second melody is also used in a paper where the researchers also used dynamic programming and an ergonomic model (Andersson & Håkansson, 2014). This research used a cost calculation that was based on previous research (Hart, Bosch, & Tsai, 2000), and (Parncutt R. , Sloboda, Clarke, Raekallio, & Desain, 1997, Vol 14) had different findings. They found an accuracy of 59.38% for melody 2 (64% in this paper). It must be noted, however, that mistakes like the first mistake in figure 11 were not made in this paper, since the researchers based their dynamic programming almost purely on ergonomics. Even though this algorithm changes fingers for the last note in the second measure, what follows is a chain of mistakes, which does not get corrected until the end of the fourth measure. Even though these fingers are still good to use ergonomically, they are not the most optimal to play with.

The big difference between results seems to be that one has to choose whether to prefer the accuracy of the fingering, which would result in a more *legato* (i.e. tied together) sounding piece or a more ergonomically pleasant fingering.

6.3. Limitations and Future Work

Due to time constraints for this research and a limited amount of previous (recent) research on generating piano fingering using different methods, there are some limitations to this paper. One of such limitations was creating a large dataset. Due to the nature of Q-Learning and time constraints, the labeling and collecting of a larger dataset were not prioritized. This could have impacted the results for training outside the test set since this would have resulted in more states. Theoretically, there should be a data set to train on that contains all data sets. This is something that future research could look into: whether an incredibly large data set would allow for more generalizability, since the agent had been trained on more data. States were based on triplets of the previous, current, and next note, which is very short-sighted. It would be interesting to know whether expanding these states (e.g. the agent looks 5 notes ahead), will give better results and if the sacrifice to generalizability is worth it.
Since this dataset was partially chosen on the differences in dynamics between pieces, this might also have impacted the results. Training on 4 Mozart pieces might have resulted in an algorithm that was better at generalizing than using 4 different composers. Future studies could look into the differences in results when using different genres. This study has only used classical pieces, which can differ significantly from more modern pieces. Features such as tempo and pauses were not taken into account but could have an impact on the results.
One of the more obvious limitations of this study was the lack of ergonomics. Future work could look into using the same algorithm in combination with a better reward scheme based on hand ergonomics to figure out if the accuracy improves, and if not, figure out why not.
Due to the lack of previous research, a comparison with other machine learning algorithms was also not discussed very intensively. For future research, it would be interesting to know whether different reinforcement learning algorithms would output comparable results and whether other machine learning algorithms (such as supervised learning) would give the same results, but also create a more generalizable model, such that training is not necessary for every new piece.

7. Conclusion

This paper has shown how reinforcement learning can generate piano fingerings based on given sheet music. Moreover, it has shown that the algorithm has an average accuracy of 77.7% for the given data. This answers the main research question as to what extent a reinforcement learning agent's labeling compares to a human-labeled one. The algorithm has shown to be very accurate, with some minor flaws. These minor flaws include hand ergonomics and a reward metric that prioritizes minimizing finger distance over the quality of the sound. However, the results have also shown that the agent understands some rules of ergonomics, without having to explicitly be taught them, which means that the agent learns not just to figure out the shortest distances, but also the most effective ways of playing them. Another goal for this research was to create an algorithm that could be used by new students to have a generated fingering, which they could use themselves. Although the algorithm has a high accuracy, it would need to be a bit

higher for humans to use it, which is something that can be done in future work. Overall, this research has shown new insights for machine learning algorithms and the way they can be used in the music learning industry, as well as provide plenty of opportunities for future research.

References

Andersson, M., & Håkansson, M. Y. (2014). *Generating Ergonomic Fingerings for Piano.* Stockholm: Kungliga Tekniska högskolan.

Eigeldinger, J.-J. (1988). *Chopin: Pianist and Teacher: As Seen by his Pupils.* Cambridge: Cambridge University Press.

English Fingering. (2001). *Grove Music Online.* Oxford University Press.

Eremenko, V., Morsi, A., Narang, J., & Serra, X. (2020). *Performance Assessment Technologies for the Support of Musical Instrument Learning.* Barcelona, Spain: Music Technology Group, Universitat Pompeu Fabra.

G., H., & Sejnowski, T. (1999). *Unsupervised Learning: Foundations of Neural Computation.* MIT Press.

Hart, M., Bosch, R., & Tsai, E. (2000). Finding Optimal Piano Fingerings. *The UMAP Journal*, 167-177.

Kaelbing, L., Littman, M., & Moore, A. (1996). Reinforcement Learning: A Survey. *Journal of Artificial Intelligence Research*, 237-285.

MuseScore. (2021, June 23). Retrieved from MuseScore: https://musescore.org/

Nakamura, E., Saito, Y., & Yoshii, K. (2020). *Statistical Learning and Estimation of Piano Fingering.* Chiba: Kisarazu College.

Nellåker, E., & Lu, X. (2014). *Optimal piano fingering for simple melodies.* Stockholm: Kungliga Tekniska högskolan.

OpenAI. (2021, June 20). *Gym: A toolkit for developing and comparing reinforcement learning algorithms.* Retrieved from OpenAI: https://gym.openai.com/

Parncutt, R., Sloboda, J., & Clarke, E. (2011). *Interdependence of Right and Left Hands in Sight-Read, Written, and Rehearsed Fingerings of Parallel Melodic Piano Music.* Australian Psychological Society.

Parncutt, R., Sloboda, J., Clarke, E., Raekallio, M., & Desain, P. (1997, Vol 14). An ergonomic model of keyboard fingering for melodic fragments. *Music Perception*, 341-382.

Rendel, D. M. (2003). In *The Harvard Dictionary of Music: Fourth Edition* (pp. 314-315). Boston: Harvard University Press.

Royal Academy of Music. (2014). *Making Music: Teaching, Learning & Playing in the UK.* The Associated Board of the Royal Schools of Music.

Royal Philharmonic Orchestra. (2019). *A new era for orchestral music.* Royal Philharmonic Orchestra.

Russell, S., & Norvig, P. (2010). *Artificial Intelligence: A Modern Approach.* Prentice Hall.

Sloboda, J. (1974). *The Eye-Hand Span - An Approach to the Study of Sight Reading .* SAGE Social Science Collections.