Artificial Neural Networks in Musical Performance

Jakob Leben

Master's Thesis
Institute of Sonology
Royal Conservatory, The Hague
May, 2012

Acknowledgements

I would like thank:

Paul Berg, for his insightful remarks that again and again made my thinking take turns

Richard Barrett, for long talks and often helping me understand my own thoughts

Table of Contents

1. Introduction

During the course of my research at the Institute of Sonology, I became interested in employing artificial neural networks in an instrument for live improvisation. Most of the literature and examples of application of neural networks in musical performance that I discovered initially was concerning either gesture recognition for optimization of mapping to sound synthesis parameters, or machine analysis of musical performance. However, I had a different idea: I was not inspired by the use of neural networks as an aid to do the work or accomplish tasks more efficiently that could have been accomplished otherwise (e.g. by human labour), but I instead desired to expose the inner workings of a neural network through sound.

Since I saw no examples, it was not clear to me at first, how this could be done in a musical performance, and especially improvisation context. One trouble in particular was, how to approach the complexity of a computation performed by a neural network in an improvisation, were, traditionally, a performer needs to be able to act quickly and intuitively. For this reason I broadened my research towards approaches to musical performance with interactive computer systems in general.

As a result, this thesis consists of the following parts: an overview of the discourse on issues of the use of the computer in musical performance in the past couple of decades (chapter 2), followed by an introduction to the theory of artificial neural networks (chapter 3), and finally a presentation of my own attempts at implementation of a computer instrument using neural networks, and software developed for that purpose (chapter 4).

2. Computers and Musical Performance

2.1 Introduction

The pioneer of interactive music systems is widely considered to be Joel Chadabe, both according to his own account (Chadabe, 1984) and that of others (Murray-Brown et al., 2011). He began performing with his system in 1977, and filed a patent for it in 1985 (US 4,526,078). In one of his rather recent papers he still states: “The concept of an interactive instrument may, in the near future, define a new way for the public to experience music.” (Chadabe, 2002, p. 2) And he is right, for it appears that even in the discourse following the onset of the new millennium the success of novel instruments designed for musical performance is still measured to large extent in relation to conditions that apply to performance on traditional classical music instruments and compositions written for them. It appears that it is primarily the mode of listening, the reasoning about the expectations of the audience, and following from that, the strategies of composition and instrument design to target these expectations, that need to be changed in order to make room for better appreciation and consequently exploration, creativity and refinement of novel music instruments.

The practice of designing novel music instruments has well mushroomed since the early attempts of Chadabe, and the technology involved has gradually become more and more accessible. We have seen in the last decade a boom of this practice and the idea has proliferated to such an extent, that in the young generations of computer music artists it has become almost self-evident that the purpose of performing on stage involves presentation of a newly designed instrument. Despite the fact that it has become common practice, we feel that it demands an evolution in reasoning about, and a shift in perception of this activity. We will walk through some theory in the field of instrument design, that has been produced in the past, before we reach to conclusions.

2.2 Technology and Virtuosity

Schloss and Jaffe (1993) explore the potential problems that “too much” technology can create in musical performance, and propose that this invites to a reconsideration of what constitutes a musical performance at all. They recognize virtuosity as one of the significant aspects of performance, adding that this applies to the perspective of the audience, and thus avoiding the question of performer's own point of view. Acoustic instruments, as they proceed, have since beginning exhibited a one-to-one relationship between performer's gestures and the resulting sound, while the introduction of computer in live performance has detached this relationship and substituted it with the possibility of very complex mapping. In the absence of perceivable coupling between the performer's activity and music, the interaction with the instrument begins to border on magic. The question that arises is, whether “we need a perceivable cause-and-effect relationship in live performance”. At that point they conclude it is a question that still needs to be answered.

According to their account the problem emerged quite recently, as it had only become a wide possibility to perform live with a computer in real-time since a few years ago. However, in a paper published a decade later Schloss (2003) continues to expose the same problem, posed in even more severe terms, saying that with the use of computers the cause-and-effect link has effectively disappeared, and that it is *necessary* to consider the visual/corporeal aspect of the performance from the observer's point of view, or the integrity of the performance is jeopardized. While Schloss and Jaffe (1993) noted in the beginning of the paper that virtuosity is not a feature of all musical traditions, Schloss (2003) later states that it is common across all cultures that the performer is typically “doing something that the audience cannot do themselves”. Further on, he emphasizes the importance of visible effort of the performer, as an expression of commitment to their activity, be it the facial expressions of a singer, “the bulging veins in the neck of the trumpeter blasting a high C, or the sweat-drenched body of an African drummer”. Even the grimace of a rock guitar player is an example, and despite the fact that the latter may be largely ingenuine, it avoids the problem that computer instruments exhibit, as “it is still a physical instrument, so the observer can extrapolate the effort without a lapse of belief”.

Based on this reasoning we could conclude that no matter how real the apparent effort and virtuosity of the performer is, and how much it really relates to how the performer actually feels and what they believe themselves, it plays an important role in the reception of musical performance by the audience. It establishes a certain trust in the audience that the performer is really contributing with their actions to the music heard. Schloss notes that these requirements may often be considered extra-musical (Schloss, 2003), but he offers little argumentation as to why the traditional strong coupling between the gesture and the sonic result is still a relevant solution to the practice of music creativity and performance today, in the first place, and why it should always be so. He does not elaborate much theoretically on the possibility that even if the old problems apply, with a new kind of instruments there might also be new solutions necessary or even latently available, or that the conditions of music perception itself may change.

We might be more inspired by looking into Schloss' and Jaffe's examples from their own musical practice. They have collaborated in a number of performances utilizing physical controllers plugged into a computer for gesture processing or sound synthesis, among which are Wildlife (1991), featuring as a case study in the earlier paper (Schloss and Jaffe, 1993), and Suite from the Seven Wonders (1996), referred to in the later paper (Schloss, 2003). Both compositions make use of the Radio Drum, played by Schloss - a device sensing own movement in three dimensions, developed by Bob Boie and Max Matthews - and Wildlife also employs the Zeta Violin, played by Jaffe - an electric violin with a pickup and a pitch detector for each string.

In regard to Wildlife, Schloss and Jaffe describe their position in a situation in which they gave computer a considerable autonomy in generating musical material as akin to that of a conductor or a cowboy:

> In such a piece, the conductor gives signals that control the large-scale flow of the music, but without specifying the individual details. To use a more colorful analogy, the independent computer processes are like cattle that are allowed to wander over the open plains and the performer's control is that of the cowboy who reigns them in when it's time to go into the corral. (Schloss and Jaffe, 1993)

To regard their activity in such a way was a solution to the problem of “feeling the music was 'getting away from us.'” Logically, this is only a solution to their own point of view on the activity, and not that of the audience. However, this addresses the other side of the issue that they did not refer to in the initial exposition of the problem.

Another interesting issue raised in Wildlife stems from such a design of the interactivity that both performers may influence the same musical parameter in different ways. For example, both may affect the pitch of the final material, and their actions can transpose the base pitch, the pitch range, or the internal harmony of the sounds triggered by the other performer. They describe this effect as “'pulling the rug out from under' the other player”. This issue is different from the one presented in the previous paragraph in that it concerns the problem of cause-and-effect link as far as “vertical” structure or momentary result is concerned, rather than temporal structure. As opposed to the former issue where the control is lacking in regard to *when* the change in the musical material happens, here the emphasis is on lack of control in regard to *how exactly* the change happens, although the change is triggered entirely as a result of the performer's action.

Both sides of the issue, however, may well fit within the general problem of lack of cause-and-effect link, that is, lack of performer's control over different aspects of the sonic result, and unpredictability thereof. In any case, these are examples of issues experienced by the performer and not the audience. Schloss and Jaffe (1993) express that they can be overcome by extensive practice and improvisation, which allows in the first place to discover the specific difficulties in the interactivity with a complex system, and then to explore and learn possible ways to overcome them. We may ask at this point, how can unpredictability, if seen as problem in itself, possibly be overcome? They offer quite an indicative example, which for that reason I quote here in its entirety:

> As a simple example, in the fourth movement, the violinist supplies the pitches that make up the percussionist's improvisation. The percussionist can choose to play recently-played pitches or can go back in time to pitches played earlier. In this context, the violinist plays only occasionally and in such a manner as to change the flow of the ongoing music. This movement was particularly difficult for him because the effect of material he played was evident only some time later when the percussionist played these pitches. Yet, an implementation detail turned out to supply the answer. It turns out that the “remembered” pitches played by the violinist are stored in a buffer that is not circular. Thus, every hundred notes (this number was at first set arbitrarily), the buffer would be empty and would start to be refilled again.

> This quirk turned out to provide just the "foot in the door" that the violinist needed. By playing tremolo, he could fill up the whole buffer with a single pitch and constrain the percussionist to that pitch. The implementation also guaranteed that every now and then the percussionist would be forced to play only very-recently performed pitches. Thus, what started out as an arbitrary irrelevant constraint turned out to be an asset in disguise. "It's not a bug it's a feature!" (Schloss and Jaffe, 1993)

We can extrapolate from this to a conclusion that unpredictability or remoteness of the effect need not be solved by removing it. There are solutions that allow one rather to turn these features into a meaningful participant in the music. As the example above shows, despite the fact that the details of what is going to happen are unknown or can not be predicted, it is possible to learn methods of action *within* the range where control is indeed retained by the performer, possibly even actions performed *after* the unpredicted event, that establish the relation with that event, and between the event and the musical result as a whole, and thus make a meaningful place for such events in the context of the entire performance. We might go on to say that the meat of a performance with complex interactive instruments partly consists precisely in this interplay between the lack of control and the possession of it.

Returning to the problematics of reception of the performance by the audience, Schloss and Jaffe do give exemplary solutions from their practice. Both in Wildlife as well as Suite from the Seven Wonders they begin with a very simple interaction scheme, and then proceed gradually to introduce complexity. In such a way the audience has time to recognize at the beginning the basic principles of connection between the performer, the instrument and the sound, and although further on in the performance they may not understand anymore exactly how the performers do what they do, they have established by then that bond of trust with the performers, that I have written about earlier. After all, the audience may have just as little clue about how exactly a classical pianist presses the piano keys in order to produce the music, but they can enjoy it nevertheless.

We have seen that, in addition to how the problem was presented initially, there are considerable new issues brought out by the use of complex interactive instruments also on the side of the performer, not only the audience. These issues, as I have argued, may be solved in a dialectic manner and overcome in a concept of virtuosity that includes the interplay between control and the lack of it. Perhaps this is also what Schloss and Jaffe (1993) refer to in their conclusion, albeit seemingly contradictory with their initial statements on its own:

“Digital signal processing will help a great deal in this problem, because the virtuosity inherent in playing acoustic instruments can be retained”. They continue: “As for the global problem of complexity and loss of the *perception*[1] of cause-and-effect, we believe that this is a problem that must be dealt with individually in every situation, and to some extent will be answered by the response of the audience.” (Schloss and Jaffe, 1993) This is to some extent suggestive of the possibility that the mode of perception by the audience is not set in stone and may change in future, or even that it might be actively changed through the practice of composition and performance. We will return to this idea later.

2.3 Towards Exploratory Engagement

We have shown how the understanding of the issues related to performance with interactive computer instruments may be shifted and better understood if we consider that the content of the music is not only in what the performer expresses *through* the instrument and his actions, but also in the relation *between* the performer and his instrument, making this interaction itself an explicit part of the matter of musical performance. We will trace the path towards this more cybernetic understanding of the role of interactive computer instruments along the lines of difference between two papers in which David Wessel was involved (Wessel and Wright, 2002; Wessel, 2006).

Wessel and Wright (2002) introduce their subject with the following words: “When asked what musical instrument they play, there are not many computer music practitioners who would respond spontaneously with 'I play the computer.'” To set an early context for reasoning about this paper: according to my own experience and that of my colleague students at the Institue of Sonology, this is no longer true. I have seen computer music artists answering without hesitation just that, and I have given such an answer myself. This may be indicative not only of the possibility that self-perception of the young generation has changed in the decade passed since the publishing of that paper, but also that their understanding of what are the expectations of their audience has changed.

The paper is valuable, however, in so far as it examines the problems involved in the use of the computer as musical instrument, that might lead one to avoid such an answer, if they did so, and thus offers an insight into what are the computer's inherent differences from

[1] Emphasis by the author.

the classical instruments. Wessel and Wright point out that the acoustic instruments, no matter how different they are, all fall under the paradigm of "one gesture to one event". We can equate this to the characterization of acoustic instruments by Schloss and Jaffe as exhibiting a tight cause-and-effect link between the gesture and the sonic result. Akin to the metaphor of a condutor and a cowboy that the latter authors attribute to the performer, Wessel and Wright apply to the activity of the performer the metaphor of "driving" or "flying" the instrument: the performer gives directions while the instrument performs the actual movement through the musical matter. Figure 1 indicates how a generative algorithm employed in the computer as instrument allows for a much more complex mapping between gesture and sound output than is typical in an acoustic instrument.

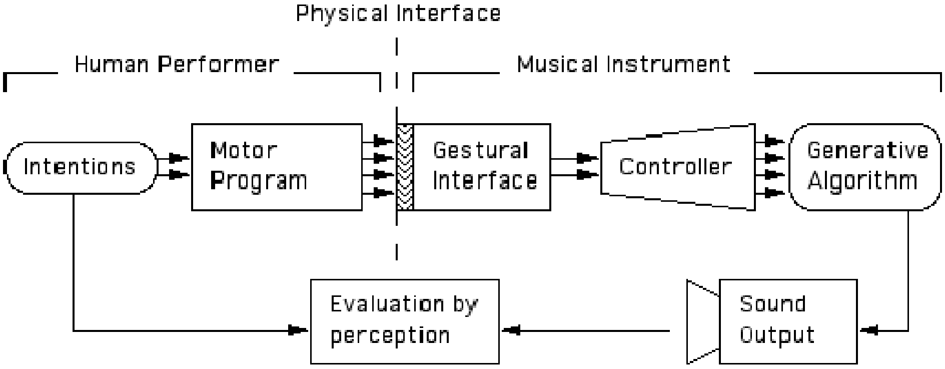

Figure 1: Conceptual framework for controller research and development (Wessel & Wright, 2002)

Furthermore, the figure bares the form of a closed loop, showing how performer's evaluation by perception of the sonic results affects their intentions and thus further action. This is true also of a performance with an acoustic instrument: despite virtuosity, momentary adjustments of gestures are performed to achieve desired sound; in addition, large-scale phrasing is often not completely determined in advance, but performed based on the perception and memory of passing musical material. Nonetheless, we may say that the aspect of feedback is more emphasized in performance where the sonic outcome of gesture can be much less expected, as a consequence of deliberate instrument design. This special attention given to the performer's preceptive evaluation of own activity is a feature shared with the science of cybernetics, and takes the same place as the feedback loop that constitutes a cybernetic system. As we will see, Wessel (2006) later expands this scheme in a way that brings it even closer to the form of systems as studied by cybernetics, and explicitly refers to theoreticians of that field.

Wessel and Wright (2002) also deal with a wider topic than Schloss and Jaffe (1993), since they "concentrate on factors such as ease of use, potential for development of skill, reactive behavior, and coherence of the cognitive model for control." In this context, they consider two attributes of musical instruments at which computers differ in comparison to acoustic instruments: namely "low entry fee" and "no ceiling on virtuosity". Traditional instruments often demand a large amount of practice before any sound at all can be made with them (consider wind instruments), or at least in order to gain any useful control of the sound output. However, they allow one to improve and expand their mastery of the instrument and attain ever more skill in musical performance. Computers on the other hand offer the possibility of design where simple instructions produce rich sonic results and typical interfaces easily available to control them are often rather toy-like and "do not invite continued musical evolution". The primary question that Wessel and Wright try to answer therefore is: "Is the low entry fee with no ceiling on virtuosity an impossible dream?".

We may ask ourselves in return, how does this relate to the problems of computer music performance that we have addressed so far. The subject of Wessel and Wright definitely concerns rather the performer than the audience, and it is justified in the light of the process of learning and expanding the skill in handling an instruments as the primary topic. But we could speculate that the "low entry fee" for the practitioner to learn also affects and implies possible solutions to audience's understanding of the relation between the performer's activity and the sonic result, and hence understanding and appreciation of the music itself. The idea of music as being grounded in the play and interaction with the instrument may be extended into the sphere of happening between the stage and the audience. We will see later how a certain "game" in this sphere (not implying necessarily the audience's interaction with the sound-producing device) may propose a fruitful shift in musical paradigm.

Meanwhile, Wessel and Wright sum up the features that allow them to affirm the possibility of both low entry fee and no ceiling on virtuosity under the conceptual umbrella of "control intimacy". They continue by describing both technical innovations as well as design metaphors employed in computer instruments that satisfy the necessary conditions. On the technical side, the first condition for control intimacy is low latency between the gestural input and the ability of the computer to respond to it. This is a rather obvious condition, and most early recognized and acknowledged by computer performers themselves. The concept of "continuous control" as opposed to "discrete events" may be a less obvious one. The MIDI protocol which has been in use for digital communication between physical controllers and the computer for several decades is based on a notion of discrete events (explicit onsets and durations) and no matter how the controller itself is designed, within the MIDI protocol the communication is obliged to conform to this notion. The newer OSC protocol is rather free from that notion, albeit still a discrete message protocol, but it allows one greater flexibility in assigning the function to each digital event send through the wire. Wessel and Wright, however, describe alternative systems that they have developed, and the speed at which these devices can process gestural control may allow it to be tightly synchronized with audio synthesis at the audio sampling rate, which results in true continuous control.

On the side of the cognitive aspects of computer instrument design Wessel and Wright describe several metaphors of control that contribute to control intimacy. A too obvious mapping between dimensions of control and musical parameters, as they say, does not allow for attractive interaction and is "musically unsatisfying, exhibiting the toy-like characteristic that does not allow for the development of virtuosity" (Wessel and Wright, 2002, p.3). Instead, they propose other mappings where control dimensions collectively contribute to a plethora of musical attributes. These approaches are rather associative of everyday activities, and include metaphors of "drag and drop", "scrubbing" and "dipping". Their examples of musical embodiment of these metaphors all involve the computer producing a relatively large portion of the perceivable variety in the sonic outcome, and while the performer keeps the power of activating the sound and silencing it at will, the interaction aside from amplitude dynamics fits well to the idea of "flying" and "driving" presented earlier.

To sum up, the concept of control intimacy, although positing low latency as the first and fundamental condition, does not imply an event-by-event association of gesture to sound output. It does, however, assure the *possibility* that where an immediate effect of control over an aspect of sound is desired (however abstract it may be) it is obtainable. Furthermore the intimacy may be interpreted not as direct closeness in formal relationship between gesture and sound elements, but rather in relation to the total experience of the instrument and the ability to recognize familiar forms of interactions (with real-world objects) in the cognition of the instrument's control interface and through its sonic feedback. Although the metaphors of interaction employed in the examples of Wessel and Wright are on the first thought only apparent to the performer and not the audience, since the physical device that they used is a touch-sensitive tablet, and the gestures are relatively minuscule from the point of view of the audience and hence hardly perceivable, I would argue that with careful exposition of the methods of interaction throughout the performance, the employment of such metaphors may very well help the audience to grasp the principles behind the performer's activity.

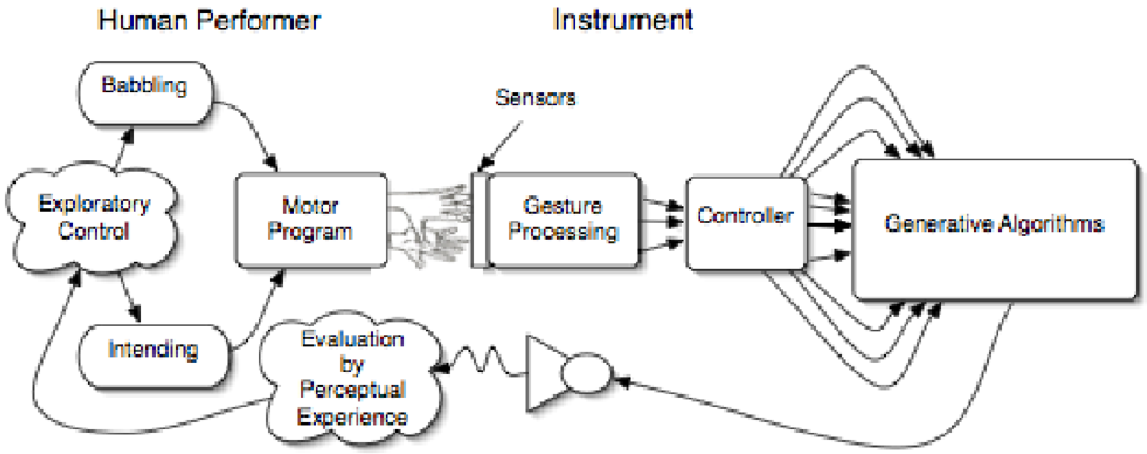

Figure 2: A framework for reasoning about computer-based musical instrumentation (Wessel, 2006)

In a later paper, Wessel (2006) extends the above mentioned scheme for reasoning about musical performance with computer instruments (Figure 1) with an addition of elements that make part in a rather exploratory interaction with an instrument, as opposed to the performance of a trained virtuoso (Figure 2). We can see that aside from determined intending, the left-most end of the structure – representing the origin of activity – features in an equivalent position the notion of “babbling”. This clear association with the way a human child learns to speak in his early experiences of the world are indicative of Wessel's research being grounded in the topic of learning and dealing with the *process* of constitution of an actor's interaction with the environment, rather than a system in which the interaction is already well established and is taking on paths well “walked in” (not to say worn out). The title of the paper alone (An Enactive Approach to Computer Music Performance) already places the research into the context of the enactive view of perception and cognition, pioneered by Francisco Varela (and others) who is considered also as one of the actors of a revolution in the field of cybernetics whereby in the 1970s the field took upon a shift in attention from how individuals steer their actions according to the feedback from the environment to considering how systems are formed, continually modified, and constituted only in the interaction with the environment. Wessel explains: “The enactive view emphasizes the role of sensory-motor engagement in musical experience” (Wessel, 2006, p.93)

By carrying out laboratory experiments analogous to those performed in research within the enactive approach to visual experience, Wessel has established that that “passive listeners do not develop perceptual skills to the same extent as those actively manipulating musical material”. Engagement in production of musical material therefore has a large effect on the development of the ability to perceive its structure and meaning. Here again, control intimacy is stated as the primary condition for successful engagement. But how does the introduction of “babbling” to the scheme of reasoning about human-computer interaction change the perspective? Wessel clarifies:

> “Babbling is distinguished from intentional commands in that auditory feedback provides information about the relationship between a gesture and a resulting sound whereas auditory feedback to intentional commands provides information about the match between the desired output sound and the actual output sound.” (Wessel, 2006, p.94)
>
> “Babbling is a non-goal-directed variation of the control parameters and is a key to the exploration of an instrument’s potential for musical expression” (Wessel, 2006, p. 97)

Babbling, then, is the process in which the gesture-to-sound mapping, that is, the relation between motoric action and the resulting sound, is processed by the human cognition through auditory feedback. Intentional action instead deals with the comparison between desire and result; the attention is all aimed towards the goal, rather than the fundamental principles of the interaction. The relationship between the performer, the instrument and the produced sound may be seen as analogous to the relationship of an individual to their environment, as studied in the field of cybernetics. It is at the stage of the individual's exploration of the environment, that is, probing, sampling the environment's response, learning the coherence of one's own actions, that a specific system of interaction is constructed. Furthermore, following a cybernetical reasoning, the system, and more importantly, the information cycling within it is constituted entirely by the process of performer's action at the stage of exploration. It is there that a unique, actor-dependent code of interaction and world of information emerge.

For the sake of communication, the individual codes of interaction may converge and find a common ground in a group of individuals – possibly to such an extent to become apparent from afar as having been predetermined, fixed, and the evolution seems to come to a halt. However, in the practice of performance with a new interactive instrument, this is rarely the case in the sphere between the performer and the audience. How, then, is communication in that sphere possible? I propose that, instead of trying to fit the performance with a new instrument into the established codes of communication, the audience is rather included in the exploratory activity of the performer and invited to witness the constitution of the particular musical world of the performance. This clarifies the aforementioned decision of Schloss and Jaffe to begin their performance with simplistic scheme of interaction with their instruments, and gradually increase the complexity; but aside from allowing for establishment of the audience's trust in the performer's ability and virtuosity, this enriches the view by explaining the principle by which the audience may come to terms with the constitution of *the information* circulating within the musical world of the performance, and hence the musical meaning.

2.4 Creativity and Constraint

When a performance with a novel musical instrument fails to be well accepted and understood, it is straightforward to blame it on the instrument. On the other hand, we have seen that the musical information may be regarded primarily as a product of action, more precisely interaction between the performer and the instrument, which may be extended, as I have suggested, to the interaction between the performer and the audience. Nevertheless, the above mentioned proponents of such a view seem to focus in larger part to the features of the instrument that enable successful interaction. Wessel's (2002, 2006) suggestions concern either the attributes that enlarge the freedom of the performer from constraints (low latency, continuous control) or make the instrument more suggestive of and intuitive in the modes of interaction.

Gurevich, Stapleton, and Marquez-Borbon (2010) have taken their research in the opposite direction. They conducted an experiment that involved 9 volunteer undergraduate and postgraduate music students, and a one-button instrument. The aim was to assess the possibility and conditions of development of individual style within the constraints of performance with this tremendously simple instrument. The participants were asked to practice with the instrument for one week, and then give a two minute performance, followed by an interview about their experience.

The experiment draws on the previous work by Gurevich, Stapleton and Bennett (2009), where the authors proposed that "within the NIME[2] discourse, this concept of style is more useful than the traditional discussion of expression, as it disentangles behaviour and action (and the perception thereof) from confounding phenomena like emotion and the construction of meaning that are implied by the term *expression*." (Gurevich et al., 2010, p.106). This is useful because it allows for a definition of virtuosity as the ability to "not only realize difficult or complex structures, but to do so with a style deemed desirable." (Gurevich et al., 2010, p.106). This implies that in order for the audience to perceive virtuosity, they must be able to separate structure from style, and Gurevich et al. (2009) argue that it is the constraints in interaction with the instrument that allow so:

> The structure of juggling–tossing multiple objects from hand to hand–facilitates the differentiation of juggling tricks (e.g. shower vs. cascade) as stylistic variations on a common structure much more readily than a significantly looser structure such as *things I can do with 3 balls (Gurevich et al.,* 2009*, p.216)*

While a more loosely constrained interaction may allow greater diversity in style, it will at the same time impede the audience's ability to recognize the variations in style, and hence virtuosity. Gurevich et al. (2009) concluded that the appropriate aim is to determine the optimum amount of constraint that facilitates both diversity in style and the ability of its recognition. In contrast, Gurevich et al. (2010) attempt to experimentally verify how diversity of style is affected at the extreme end of constraint.

Constraint of a design is defined by Gurevich et al. (2010) as "the strong indication it gives the user of a singular method of use", rather than "the number of possible actions offered by the design" (Gurevich et al., 2010) which relates to the number of affordances. Due to the particularities of the technical implementation of the instrument used in the

[2] The International Conference on New Interfaces for Musical Expression

experiment, it contained elements that increased the amount of affordances beyond that of just pressing the main button. The button would ideally turn on and off a single-pitched sound, but the pitch ended up strongly coupled to the power supplied by the battery, and would variate with the amount of charge in the battery. Moreover, the device featured a power switch, to avoid unnecessary battery use, and it was possible to affect the sound by switching the device on and off with the main button pressed. The sound could also be filtered by covering small holes in the enclosure behind which the speakers were placed in the interior of the instrument. It also makes part of the affordances for example that performance could make use of spatialization by moving the whole instrument in space. Nonetheless strong constraint was constituted by the fact that the design exhibited a very clear suggestion of its intended use which was solely to play the tone by pressing the button, and there was no suggestion by the authors of the experiment as to what other possibilities to explore. In spite of that, other aspects of the performance in addition to variations in sound output (length of notes, length of silences, etc.), were evaluated as part of the style, including different ways that the sound-triggering button was depressed (aside from an obvious press with a finger), the ways that the device was held, and the posture of the performer.

Experimenters found diversity among the participants in all of these aspects. Despite the fact that many participants expressed the initial sense of excessive simplicity and limitation of the instrument, they were eager to explore the possibilities and discover uses of the instrument beyond the simple one primarily indicated. Based on the results and the participants' reports of their practice with the instruments, Gurevich et al. (2010) identify two distinct types of exploration: a vertical and a horizontal approach. The participants that fall into the class of vertical explorers would attempt to exhaust the possibilities within a single affordance, and then discover and explore a different one only as a result of meeting difficulties, and identifying problems underpinning dissatisfaction with the musical results of their earlier approach. The horizontal approach, on the other hand, denotes an attempt at enumeration of all the affordances before exhausting the possibilities within each, and the drive to search for another way of interaction with the instrument as soon as one is discovered. Despite the difference between the two participants that strongly exemplified each of these distinct approaches, they both shared an extremely exploratory attitude, and would rate their own performance with lowest grades within the group, implying the belief that there is still a lot more to explore and master in relation to the instrument.

Gurevich et al. (2010) conclude that diversity in style of performance with an extremely constrained instrument was achieved by the performers both in spite of and because of the constrained design. It was precisely the constraints that enticed exploration of possibilities beyond the immediately suggested affordances. On the other hand, as noted later in retrospection by Marquez-Borbon, Gurevich, Fyans and Stapleton (2011), exploration was possible also because there was no suggestion or indication given to the participants in the experiment as to how they could or should use the instrument. Although the design of the instrument itself strongly suggested a particular use, there was a kind of freedom in the lack of external prescription. Can we therefore still characterize the conditions surrounding the performances in the experiment as constrained? And how does this relate to the problem of the vanishing of the cause-and-effect link in computer instrument performance due to the use of complex gesture-to-sound mapping, that, we may say, constitutes another kinds of freedom and constraint?

The freedom of exploration in the one-button instrument experiment constituted in the constraint of suggestiveness, both external as well as internal to the instrument. Although the instrument exhibited an apparent strong indication of use, this may be so only because of the absence of a broader spectrum of indication – when the spectrum narrows, the relative amplitude of its contents seem relatively boosted. In contrast, a broad spectrum of suggestiveness may be paralleled to noise. Contrary to possible straightforward conclusions, the narrow spectrum contains more valuable information to support exploration: the absence of immediately suggested possibilities can just as well play a role of positive information. Marquez-Borbon et al. (2011) associate positive support for application of skill by experienced performers with ambiguity in instrument design. It is important to recognize that ambiguity in this sense relates to the presence of minimal information in a vast field of openness, rather than no information, or saturation with information. A field saturated with possibilities is equal to a field of emptiness, with no point of departure and hence no possibility of movement. Ambiguity on the other hand is ambiguity about *something*, ambiguity about what path to take from a particular *given* standpoint. This suggests that richness of interactive instruments, allowing ultimately development of virtuosity with personal style, should be based on such kind of ambiguity rather than a large quantity of complex pre-designed and suggested modes of interaction.

Nevertheless, we should hope that the bottom line of the above propositions is not that it is best for new instruments to be simplified to the level of the one-button instrument used in the experiment by Gurevich et al. (2010). The availability of computers and contemporary technology naturally drives the desire for performance with instruments of more complex interactive schemes. We have already seen earlier in relation to the examples given by Schloss and Jaffe (1993), Schloss (2003), Wessel and Wright (2002), and Wessel (2006) suggestions that enable one to attain virtuosity with instruments that differ in their interaction from traditional one-to-one, event-by-event relationship between gesture and sound in acoustic intruments. The concept of ambiguity in design presents further elaboration on how the interaction scheme should be structured. Most importantly, it adds to the idea of importance of support for exploratory activity that we have touched upon before in exposition of a cybernetic approach to musical performance, and gives a clarification of how the *dynamics* of this activity work, and on what kind of features of an instrument they may be based. We may conclude: no matter how large amount of affordances an instrument ultimately offers, and how complex relation between gesture and sonic result they may allow (or precisely due to these particularities) - in order to support development of virtuosity, which includes the development of personal style, the instrument shall best bare a certain set of constraints in its *interface* that constitute a productive ambiguity and invite for a process of discovery in which a musical vocabulary is constructed through engagement, instead of being prescribed.

However, as far as the perception of virtuosity and understand of a musical performance by the audience is concerned, Marquez-Borbon et al. (2011) conducted further experiments proving that the audience (including experts) can get into trouble understanding how the instrument works, i.e. the gesture-to-sound mapping, even by introduction of very little complexity:

> Among some participants, this perceived complexity led to inflated evaluations of the performer's skill and experience with the instrument. Other participants thought the performer's actions amounted to mere "button-pressing," suggesting the instrument was simple to master. Many concluded the performer must therefore possess intimate technical knowledge of the instrument, rather than bodily skill, in order to produce such a rich variety of sound. There was a similar perception that the performer was not fully in control of the sonic output, that he simply mediated some aspect of an automated system. Without an accurate understanding of the interaction many

> spectators found it difficult to assess attributes of the performance such as skill or error. (Marquez-Borbon et al., 2011, p. 375)

The aim of the experiment was precisely to asses the ability of the audience to understand the performer-instrument interaction when confronted with performances with instruments of different degrees of familiarity. The instrument involved in the performance referred to in the quotation above was deliberately designed to be completely unfamiliar to the audience and would provoke a difficulty of understanding. However, no deliberate attention to the way the performance was structured is mentioned.

In contrast, I have suggested that the problem exposed in this experiment might be addressed by including the audience in the exploratory activity of the performer. I could now elaborate on this suggestion: an instrument designed for the kind of productive ambiguity described above, in such a way as to invite the performer into a process of creation of a unique mode of interaction through the process of exploration and discovery, offers the possibility of re-playing and enacting the same process during the performance. Specifically, the movement through the evolution of interaction with the instrument and thus musical material may be more accessible to the audience when it consists of a dialectic in the field of "real" tensions between constraint and overcoming thereof, as experienced by the performer.

2.5 Composing the Instrument

An approach in this direction is proposed by Murray-Brown, Mainstone, Bryan-Kinns and Plumbley (2011). They begin by describing the contemporary problem in computer instrument performance in familiar terms:

> " Previous research points to a failure to balance complexity with usability, and a loss of transparency due to the detachment of the controller and sound generator. The issue is often exacerbated by an audience's lack of prior exposure to the instrument and its workings." (Murray-Brown et al., 2011, p. 56)

The problem can be very easily equated to that described by Schloss and Jaffe (1993) two decades ago and it seems that the same issues acknowledged at least at that time persist to this day. However, they follow immediately with a true Hegelian twist, exposing what appears to be the paradox underpinning the conflict: novel musical instruments are “intended to be both a tool for creative expression and a creative work of art in themselves, resulting in incompatible requirements.” (Murray-Brown et al., 2011, p. 56) Instead, they propose to consider the instrument, the composition and the performance as one whole. They recognize an early effort in this direction – that of Chadabe in his concept of interactive composition. However, although Chadabe's algorithm generating novel musical material on-the-fly at the time of performance may be considered both as an instrument and a composition, it was regarded as a static *tool*, intended for the musical *expression* of the author. Murray-Brown et al., on the other hand, emphasize the role of instrument design and exposition as the content of the performance in itself.

Murray-Brown et al. lay out the conditions of engagement in the performance, both that of the performer and that of the audience. On the assumption that provoking a positive musical appreciation consists in large part in appropriate management of the expectation throughout the composition (balancing with each musical event the novelty in material and its consistency with what has passed), it is possible to apply to musical performance the theory of *flow* by Csikszentmihaly (cited by Murray-Brown et al., 2011): the condition for enjoyment of music lies in a balance between the success of prediction of following events on one hand and the element of surprise on the other hand. This applies, in an analogous fashion, to the activity of the performer: pleasure in performing comes with appropriate balance between the effort and the ability to tackle, or tame if you will, the tasks imposed by the musical material and the mode of interaction with the instrument. This state of flow is important both for the performer and the audience. However, the performance with an instrument unfamiliar to the audience poses a potential obstacle to their flow in two ways. Firstly, the enjoyment of the audience stems partially from recognizing the flow of the performer (which is analogous to virtuosity), which may be hindered by inability to recognize the difficulties with which the performer is confronted, and hence also whether or not the performer succeeded in his “tasks”. And secondly, since a novel instrument often also brings unfamiliar possibilities in musical material, the ability of the audience to construct a model the musical structure on the background of which to build the expectations is largely overpowered by the difficulty thereof.

However, these issues are largely based on the premise that the instrument is intended as a *tool* for the music as an *entity apart* that needs to be expressed. An opposite approach that allows to address the issue in a more fruitful way is to consider the interaction with the instrument, even the instrument itself, as a central part of the musical performance. In this context, music is not composed *for* the instrument, but the instrument is composed. Both the relationship between the performer and the instrument as well as the audience and the instrument *develop* throughout the performance. Murray-Brown et al. acknowledge that attention to the way the instrument and its principles of interaction are exposed throughout the performance has already been present, both in theory and in practice (of which an example we have seen in the performances of Schloss and Jaffe), but what is novel is the idea that the exposition of the instrument may be seen as a temporal art form on its own.

We have already met the proposition to regard the issues of interaction with novel musical instruments apart from the notion of expression in the research of Gurevich et al. (2009, 2010) where the purpose of this was to allow better to address the conditions for the performer's development and the audience's perception of personal style and virtuosity. While the reason in that case was to separate the notion of virtuosity from that of expression and meaning, it is not the case in the intetion of Murray-Brown et al. (2011) to disregard the notion of expression *as such*. Rather, they reject the usefulness of the notion of expression as that of something *apart* from the exposition of the musical instrument. Instead, what they propose is that the temporal development of interaction between the performer, the instrument and the audience is an actual musical expression.

Regardless of this distinction the research of Gurevich et al. (2009, 2010), Marquez-Borbon et al. (2011) and Murray-Brown et al. (2011) all meet the common ground at the importance of constraint in relation to creativity and exploration. Murray-Brown et al. reason that an apparent limitation of an instrument designed specifically for a performance is not a limitation indeed, but rather an asset:

> …it is precisely within such a tightly constrained domain that new ideas happen, new ways of using (and abusing) an instrument are found, and new compositions, or even new types of music, are created. In a time when musical programming languages have unleashed a bewildering amount of sonic potential, it is the constraints rather than the affordances of an instrument that characterise it. (Murray-Brown et al., 2011, p.58)

In the context of their approach to the instrument design, the composition and the performance, Murray-Brown et al. suggest the following three design principles:

1. Design for a single performance
2. Consider the rate that structures emerge
3. It is easier to begin 'in the dark'

The first principle perhaps introduces most of the novelty in approach, as it dictates focus on *mutual* influence of music and instrument at their creation, presentation, consideration of impact on the audience, as well as the narrative of the performance. We have seen an employment of the second principle already in performances by Schloss and Jaffe (1993, 2002) in the gradual introduction of complexity of interaction with the instrument in the course of performance, but Murray-Brown et al. place an emphasis on how the development of the interaction affects and constitutes the narrative of the performance and finally the totality of the music. The sound material should not be neglected as being only introductory and demonstrative until the complexity of interaction reaches a state at which the "expression of music" would be possible, but the music should be regarded as already happening in the full meaning of the word since the beginning of development, and with full responsibility for and contribution to the aesthetic value of the performance. Finally, the third principle simply states how precious we may consider an audience that has no knowledge of what will happen in the performance, and hence complete openness of expectation.

As a demonstration of practical application of the three principles, Murray-Brown et al. (2011) continue by giving an example of an instrument and a performance named The Serendiptichord, created in collaboration of the first two authors, Murray-Brown and Mainstone. The Serendiptichord is a wearable instrument for dancers, embodying the following ideas: "exploration, discovery, serendipity, inspiring creative movement and provoking playful behaviour". We may notice here an echo of the features attributed to the activity of "babbling", as described by Wessel (2006) within the enactive approach to computer music performance. The actual course of the dancer's familiarization with the instrument was reflected in the composition of the performance and the interaction with the audience:

> As the instrument was developed, a narrative emerged of the relationship between performer and instrument through stages of discovering the instrument […] The

> narrative not only serves to unify music, instrument and interaction: it provides a framework for the instrument to be communicated to the audience. […] Establishing transparency – the connection between audience and instrument – is part of the aesthetic experience. (Murray-Brown et al. 2011, p. 58)

2.6 Conclusion

I have attempted to show a path through argumentation in support of a view that a shift in reasoning about the computer music performance may place the discourse about the issues related to performance with interactive computer instruments under a better light, and into a more fruitful soil. The problems of the performer's development and the audience's perception of virtuosity in relation to interactivity consisting of a complex gesture-to-sound mapping can find a solution by changing the perspective both on what constitutes the virtuosity and what constitutes the performance. Based on examples from compositional and performative practice of Schloss and Jaffe (Schloss and Jaffe, 1993; Schloss, 2003), we have established that virtuosity may be developed even in the absence of predictability of the exact sonic output of the instrument, albeit a virtuosity that consists in an interplay between possession of control and the lack of it. An enactive approach to computer music performance (Wessel, 2006), inspired by the theory of cybernetics, further shifts the focus of attention to the *process* of construction of musical meaning through exploratory engagement in music-making with the instrument, and suggests the inclusion of the audience in witnessing of the becoming of a musical world through performance. An examination of the effects of particular kinds of constraint in the design of the instrument (Gurevich et al., 2009; Gurevich et al., 2010; Marquez-Borbon et al., 2011) led us to the concept of productive ambiguity, which characterizes the design *for* exploration. Finally, the concept of instrument composition summarizes the above propositions into an approach that regards the exposition of the instrument and its methods of interaction and sound production as a temporal art form on its own.

I consider the latter as an extension of the principles of constraint as applied to instrument design into the principles of composition. In my view, the principles of constraint in instrument design establish an inherent dynamic in the instrument in regard to the way that the interaction by an engaged musician unfolds in time and with exploration. When such an instrument is “composed”, i.e. designed specifically for a performance and considered as one whole with the composition of music, the same inherent dynamics of the instrument come to work in the sphere of interaction with the audience, and allow for it to be engaged in the logic of unfolding end evolution of the music.

3. Artificial Neural Networks and Application in Music

Artificial neural networks are a mathematical or computational model inspired by the structure and functioning of the real neural networks that make up the biological brain. They have been developed mainly with the purpose of making possible or more efficient the computational solution of certain problems,.by taking advantage of the increasing knowledge of the role, structure and functioning of our own brain. In other words, the aim is to enabled the computer to do specific kind of work *instead* of us, with the same (or better, super-brain) efficiency as we do – the kind of work that it couldn't perform using more traditional models of computation employed before, although they already allowed it to excel over our capabilities at other kinds of computation and logical analysis. I am referring for example to recognition of patterns and comparison of complex data by similarity without prior knowledge of the structure of the data. This type of work is most commonly considered to play an important role in what we understand by *thinking and perception*, and hence neural networks are often employed in the computational models that power artificial intelligence, and used in robotics. Neural networks are very useful in a number of scenarios for the reason that, by defining an underlying abstract model of computation - in independence from the data to be processed - they can be applied to a wide range of problems by means of self-adjustment of own internals in accord to structures inherent in the input data, without the need of our intervention, and hence knowledge of those inherent structures.

Since the development of artificial neural networks is one of the propellants of the possibility of artificial subjects, autonomous in activities equal to those that a human can participate in, they have naturally been exploited also in the field of music, ranging from the activity of music analysis to music making and performance. Often, they are particularly useful for their autonomy of operation, precisely because they liberate us from the need to *understand* the information that they process; their internal operation is itself regarded as understanding, whereas we treat them as a black box and we are only interested in the *facts* that they produce as a result of processing the information. I, however, am most interested in the use of neural networks as an aid in understanding of the process by which the information is constituted, processed and exchanged by ourselves, including the information and its exchange that makes up the world of musical communication and expression. By regarding a neural network system as a mirror, rather than a substitution of our own activity, and engaging in an exploration of its internal behavior - exploration in the same sense that I exposed in the previous chapter in relation to interactivity in computer instruments – we can come to a reflection of our own understanding of music.

For the purpose of the topic, I will continue this chapter with an overview of the fundamentals of the theory of artificial neural networks and their various types and modes of application, after which I will present several paradigmatic examples of employment of neural networks in the field of music, computer instrument design, and interactive music performance. I will refrain from the use of formalized mathematical expressions and rather use literal descriptions, as I think it is appropriate in this rather musicological context. The information regarding neural networks may be verified by consulting Kriesel (2007), whenever not specified explicitly.

3.1 Artificial Neural Network Theory

3.1.1 The Brain and the Computer

There are some problems that we meet in our daily life and are quite easy for us to solve instantly, but are rather hard, if not impossible, to explain exactly how we do it. For example, recognizing a friend from a large distance, or fitting an unknown song into a particular genre after hearing just a bar of music. We would run into a great trouble trying to put down a series of instructions for a computer to follow, in order to do the same. How do we learn to do it then? The answer is already there: we *learn*. The knowledge required to solve the problem becomes embedded in our brain, without our consciousness of an explicit recipe for solution. Computers, however, can not work without such recipes. The computer, as a Von Neumann machine, operates in sequential steps, each step performing a well-defined instruction. In contrast, neurons in our brain all work at the same time – in *parallel*. As knowledge is gained, it becomes embedded in the developing structure of our brain – it is *distributed* across the whole collection of neurons.

This does not have effect only on possibilities of computation, but also on speed. Although the computer contains a comparable number of transistors (the smallest elements of a digital processor) to that of the neurons in our brain, and the speed at which a transistor can change state is several orders of magnitude faster than that of a neuron, a computer still does not compare in performance to our brain at several tasks, due to the sequential nature of its operation.

Another distinction that the distributed nature of the brain brings is in regard to f*ault tolerance*: a damage at a part of the brain does not stop the operation of the whole. A computer program, however, crashes as soon as an invalid instruction is met, because it is undefined in that case what the next instruction should be. Moreover, since the brain is *plastic,* meaning that the connections between neurons can be reconfigured, and are continuously being reconfigured as we learn, a fault in part of the structure can be remedied by re-establishing the same functionality by modification of other parts. But the brain is not tolerant only in regard to internal errors, but also external errors: we can understand speech, or recognize the speaker even if the voice is heavily affected by various influences and departs from the usual.

Fault tolerance and the ability to learn is tightly connected with the possibility of *generalization* of knowledge and *association* of data: by observing several instances of songs within the same genre, we can generalize to the abstract common properties, and are able to associate further instances with the same genre.

Nonetheless, the computer, as a Turing machine, has the capability to simulate any other computational machine, so the idea behind neural networks is that a simulation of the parallel architecture of the brain on the computer will allow to implement the same features:

- self organization and learning
- generalization and association
- fault tolerance

3.1.2 History

The first attempts at the computational modeling of the brain date back to 1943 and were carried out by Warren McCulloh and Walter Pitts. The development has gradually accelerated since then, until Marvin Minsky and Seymour Papert published a paper in 1969 disproving in a rigorous mathematical way the usefulness of the models available at that time for solving many important problems and predicted that the research was destined to meet a dead end. Despite the wide unpopularity of the topic that this caused, the research was picked up in 1972 by Teuvo Kohonen (also known for invention of the self-organizing map – a kind of neural network) and slowly reinvigorated until John Hopfield (the inventor of the Hopfield networks) in 1985 and the Parallel Distributed Processing Group in 1986 found acceptable solutions to the problems described by Minsky, and launched the field into an era of explosive development which lasts until today.

3.1.3 Structure

An artificial neural network consists of a set of *neurons*, and *weighted connections* between them. A connection supplies the data outputted by a neuron as an input to another neuron, while the weight has an excitatory or inhibitory role on the data - it amplifies or reduces the effect of the data on the neuron at the end side of the connection. In computational terms: the data processed by neurons are numbers, which while transferred along the connections to other neurons are multiplied by the weight associated with a particular connection.

A neuron processes the incoming data in several stages: the first stage is expressed by a *propagation function* which translates the input data vector (of which elements are data coming from outside the neural network, or other neurons) into a scalar value. Most commonly the propagation function is simply a weighted sum of the input elements (hence the propagation function includes the implementation of connection weights).

The second stage is defined by the activation function, which describes the neuron's activity (output) in relation to the network input produced in the first stage. It implements the "switching behavior" of biological neurons: the latter exhibit a threshold in relation to the amount of electronic activity on the input side, above which they start to "fire" - produce electronic activity on their output side. The state of inactivity translates in an artificial neural network into a low output value, and the state activity into a high output value. Examples of commonly used activation functions are:

- *binary function*, sometimes also called *Heaviside* function: produces either a 0 or a 1, according to whether the input is above or below the threshold
- *hyperbolic tangent*: for input values substantially below the threshold the output converges to -1, and for those substantially above to 1, while input in the range close to the threshold is mapped to a non-linear continuous slope between -1 and 1
- *Fermi function*, also named *logistic function*: similar to hyperbolic tangent in shape, however its output is in the range of 0 to 1, and it can be expanded with the "temperature" parameter that defines the steepness of the slope in transition between the extremes

And a less common activation function:

- stochastic: output depends on the input according to a random distribution

The third stage of data processing within a neuron is the output function, which can be used to further process the data before it is transferred to other neurons, although very commonly this is simply an identity function, which is equivalent to omitting this stage.

Neural networks can be designed with various topologies, that is, the ways that neurons are connected. We distinguish three types of topologies: *feedforward*, *recurrent* and *completely linked*.

A feedforward network consists of layers of neurons, and each neuron can only have connections to neurons in the next layer. The consecutive layers are named according to their role: an input layer (the group of neurons processing data from outside the network), several hidden layers (invisible from outside), and the output layer (the neurons outputting the final result of the network processing). A variation is a feedforward network with *shortcut connections*: connections are not only allowed to the next layer, but also to any subsequent layer, and hence such connections are called *shortcuts*.

A recurrent network is one in which neurons may affect their own activity either directly (*direct recurrence*), by having connected their own output to their input, or indirectly (*indirect recurrence*) if connections towards the first layer are allowed, so that a closed loop may be formed through a series of connections. Such networks do not always have explicitly defined input or output neurons.

In completely linked networks, connections between any pairs of neurons are allowed, except for direct recurrences. Moreover, the connections are required to be symmetric. There is no notion of layers, so any neuron can be an input or an output neuron to the network as a whole. A popular example of such a network is the self-organizing map, also called Kohonen map after the name of its inventor.

Training a neural network to produce a specific output in relation to input (analogous to learning in humans) consists of modifying the structure of the network (adding and removing neurons and connections), the connection weights, the types of activation functions or the thresholds of neurons. However, to ease the training of thresholds, a common approach to implementation is to represent the thresholds within the structure instead of within the neurons - as weights of connections from an additional neuron with a constant output (typically 1). Such additional neurons are therefore called *bias neurons*. This way, training that only consists of modification of weights can also affect the thresholds.

According to the way that the computation within a neural network is performed, approaches can be divided mainly into two categories: *synchronous* and *asynchronous* activation of neurons. Synchronous activation is closest to the biological neurons – each cycle of computation consists of each neuron calculating its network input by means of the propagation function, activation by means of the activation function, and output by means of the output function. Synchronous activation can be used on any kind of topology, but it is not always useful, and thus computationally inefficient in many cases; consider a feedforward network with N layers: it would take N times the calculation of *all* neurons in order for the input neurons to have an effect on the output neurons.

On the other hand, asynchronous activation means that the neurons do not calculate all at the same time. Variations include *random order* of activation (a cycle for a networks of N neurons consists of N times randomly choosing a neuron for calculation), or sometimes more useful *random permutation* (each neuron is activated exactly once within a cycle, albeit in a random order). A variation reasonable for feedforward networks is *topological order* – the order is chosen according to topology, for example from the input layer through consecutive hidden layers, to the output layer. Topological order is only applicable to non-recurrent networks, though. In most cases, however, it is useful enough and most efficient to predefine a *fixed order* instead of determining it again at every cycle; although some networks can change their topology and hence a fixed order may not be useful in that case.

Input and output dimensions of a neural network signify how many neurons receive data from outside the neural network (the *input neurons*), and how many neurons are considered to produce the result of the network's total computation (the *output neurons*). The sets of data given to the input neurons and data produced by the output neurons at each cycle are called *input and output vectors,* respectively.

3.1.4 Training

As explained earlier, the most interesting feature of neural networks is to be able to learn appropriate response (output) from examples (input), and then to arrive at generalized rules which it can apply to any new data not seen at training. The data supplied to a neural network at training is called a *training pattern*, and the set of all training patterns – a *training set*. As we have seen, a network learns by modifying its structure, which may include adding new neurons and connections, removing the old, changing connection weights, neuron thresholds or neuron functions. But except for the latter, these features can be to some extent simulated only by the change of connection weights, and thus the training is usually an algorithm that prescribes how the weights should be changed in relation to the training patterns.

Paradigms of training and application of neural networks may be classified into three large categories: *unsupervised learning*, *reinforcement learning*, and *supervised learning*. In unsupervised learning the training set only consist of input vectors (*input patterns*), and the network tries automatically to identify similar patterns among those, and classify them into similar categories. In reinforcement learning, the network processes the training set consisting of input patterns, after which it is given the information whether it has performed well, and possibly how well, which it uses for learning. In supervised learning, a training pattern consists of two vectors: *training input* is a vector of input values, and *teaching input* is a vector of correct output values for the training input. Thus, for each processing of a teaching input, the network can compute the exact *error vector*, which is the difference between the actual output and the teaching input, and hence also called *difference vector*.

It is particularly of interest to know whether the network has only *memorized* the input data, or it has actually learned the inherent rules, and is able to generalize to new data after training. In the former case, it would produce correct output for all the training patterns, but fail to do so for any other pattern. This may be a result of *overtraining*. It is therefore useful during reinforcement and supervised learning to test the network error both on the training set (which contributes to connection weight changes), and a separate *verification set*. The training should stop when the network provides good results for both sets; if the error is still dropping for the training set, while it starts to rise for the verification set, that is an indication of overtraining.

In the continuation, I will present several popular paradigms in the field of neural networks.

3.1.5 The Perceptron

One of the most popular paradigms among those of supervised learning is the *perceptron* – it is a feedforward neural network with (possibly) shortcuts connections. In analogy to the morphology of the eye, this network has a layer of neurons for data acquisition, named *retina*, with statically weighted connections to the next layer of neurons with identity as activation function, followed by one or more layers with trainable weights and different activation functions. Since the second layer only collects the information from the retina (does not process it, due to the identity activation function) and the weights between that layer and the retina are not trainable, it is the second layer that is considered the input layer. A *singlelayer* perceptron thus contains, aside from the retina, only an input and an output layer, which results in one layer of trainable weights and "switching" activation functions. A *multilayer* perceptron contains additional (trainable) hidden layers.

The error of the network is seen as a function of weights and the differences between the teaching input and the network output. It is possible to derive this function, based on the activation functions, so the training consists in modifying the weights in the direction that minimizes the error function. In the multilayer perceptron, a training procedure called *backpropagation of error* is applied: first, the weights of connections to the output layer are modified in the same fashion as in the singlelayer perceptron, but the error of the network also affects the weights of connections to the preceding layers, as they are modified according to the change applied to the weights of connections in the succeeding layer.

The perceptron is used as a function approximator: it has been proven that a singlelayer perceptron can arbitrarily precisely approximate any linear function, while a multilayer perceptron can do the same for any function, and is thus considered a *universal function approximator*. In other words: if there is a function that defines a dependence between one set of data and another set of data, a perceptron can be trained to "discover" that function on several examples of related pairs, and then provide a matching pair to an input data according to the learned dependency. Moreover, in regard to fault tolerance: even if an input pattern is inconsistent with itself (the elements are not in such internal relationships as exhibited by the training patterns), the network will output a pattern that would match a valid input pattern most similar to the actual one.

3.1.6 Recurrent Neural Networks

Recurrent neural networks may also be a subject to supervised learning. Within this class of networks there is a great number of paradigms, but some examples may be *Jordan* and *Elman* networks. Jordan network is a multilayer perceptron with an additional set of *context neurons*. These are neurons that each receive information from one of the output neurons, and feed it back to all the input neurons. Elman networks are similar, but they have one layer of context neurons for each information processing layer; the role of context neurons here is analogous to those of Jodan networks: each neuron in a processing layer is connected to one neuron of the corresponding context layer, while each of the context neurons is in turn connected to all the neurons in the corresponding processing layer.

Such recurrent multilayer perceptrons contain internal state, which is said to store the context of the network as a whole (in case of Jordan networks) or the context of each separate layer (in the case of Elman networks). A recurrent network can compute more than a feedforward network: if the recurrent weights were set to 0, the network would be reduced to an ordinary multilayer perceptron. Since the network can access its own context of processing, its structure embeds a notion of time, and hence recurrent networks have been applied often to problems of time-series prediction (predicting the next value based on the previous value). However, feedforward networks can also be used successfully for this type of problems in the following paradigm: each input vector contains a moving window of several past values in succession, while the output is the next value to follow. Comparative studies have shown that a feedforward network can sometimes perform even better in time series prediction (Hallas and Dorffner, 1998), and are often used for simplicity of training.

It is more complicated to train recurrent neural networks due to internal loops. One common technique is *unfolding in time*, whereby the connections to context neurons are broken, and a copy of the network representing its previous state is attached on top of the context neurons (the output neurons of the copy become the context neurons). This procedure is repeated arbitrarily many times, to account for the effect on the network of a desired amount of recurrencies. Effectively, we obtain a feedforwad network, which can then be trained with any algorithm. However, this greatly increases the computational cost of training. Alternative methods without a limit on the recurrence span include *recurrent backpropagation*, derived using differential calculus, and evolving the network using evolutionary algorithms.

3.1.7 Hopfield Networks

John Hopfield developed these neural networks in inspiration by magnetism: particles move and rotate in a magnetic field in such a way as to reach an energetically most favorable condition, i.e. the lowest point of the energy function in relation to magnetic charges of other particles. In a neural network, the spin of the particles translates well into neuron activation state, and the state of energetic equilibrium into the state of minimized error. A Hopfield network is thus a completely linked network, where each neuron affects the activation of another neuron symmetrically. Complex dynamics can be achieved with simple binary activation functions. However, despite the complex dynamics, it has been proven that a Hopfield network will eventually come to a standstill. Thus, the input to the network is considered to be the initial activation to which the neurons are initialized, and the output the state of neurons after the network stops.

Since the neurons affect each other symmetrically, a positive weight on the connection between two neurons will force them towards the same activation state, while a negative weight towards the opposite states. A weight of 0 signifies no effect on activation. Since input and output patterns are represented by neuron activation, the training consists of comparing all the pairs of values within a pattern, and adjusting the weight of the connection between corresponding neurons in the following manner: if the input values are the same, the weight is increased, otherwise it is decreased. This is repeated once for each training pattern. In the end, a weight is high if the two corresponding input values were the same for many patterns. Effectively, when the network is trained, a training pattern on the input will result in the same patterns returned as output (the network it will stop immediately). For any other input pattern, the closest matching training pattern will be returned.

This is called *autoassociation*, and hence autoassociative networks are primarily used for pattern recognition and pattern completion. Conversely, by reversing the rule for updating the weights during training (decreasing weights for equal input pairs and increasing for opposing pairs) a Hopfield network can be trained for *heteroassociation*. This way, the network will associate one pattern with another one, which works similarly to human memory where a particular experience can trigger a memory of another.

3.1.8 Self-organizing Feature Maps

Self-organizing features maps or SOMs, developed by Teuvo Kohonen in the Eighties, are similar to Hopfield networks in that the output is the state of the network, but they can learn completely unsupervised. They are used to map a high dimensional input space into a topology of less dimensional space. For example, they can sort highly dimensional instances of input data into a line or a two dimensional grid according to similarity; similar data will be mapped to a position closer together than dissimilar.

It is possible to describe the structure of SOMs in already presented terms from the field of the neural networks, but a clearer description of their functionality may be given in slightly different terms. As said, the neurons are arranged into a low dimensional topology like a line or a grid, with equal spacing. Each neuron, however, is associated with a value of the same dimension as the input space, and that value is named *neuron center*. For the purpose of training, the neuron centers are initialized with random values. When an input value is presented, a neuron is searched for which has its center closest to the input value – the *winner neuron* – and this neuron is said to be *activated*. If several neurons are in a tie, it does not matter which one is picked. The difference between the input and the winner neuron's center is then calculated, and added to the center of the winner *and* the neurons in its topological neighborhood, in proportion to their topological distance from the winning neuron – the closer neurons are pushed more towards the input value than the more distant ones. The SOM learning is unsupervised in that there is no notion of error; the more samples from the input space it is presented, the better its neuron centers will represent a topological arrangement of the input space by similarity.

Finally, after the networked has learned enough, what is interesting is to present it with a new input and see *which* neuron is activated, rather than what is that neuron's value. We then know that the input values that previously activated this neuron, or the neurons in its neighborhood, are similar. Naturally, we have to had somehow stored the previous items from the input space together with the information about which neuron they activated, and hence SOMs are typically used for associative data storage. This allows for context-based search where an input item retrieves items from the same or similar contexts.

3.2 Application of Artificial Neural Networks in Music

There is many fields related to sound and music where artificial neural networks are applied, including for example speech recognition, gesture recognition, music classification, musical score analysis, automated music performance, etc... I will present several examples of application that, in my view, are both particularly relevant to music performance as well as paradigmatic. They are mostly related to gesture recognition.

3.2.1 Early Examples

One of the early reports about application of artificial neural networks in gesture recognition is by Lee, Freed and Wessel (1991). They developed an object for neural network computation, for the MAX graphical programming language, which allowed for easily conduct experiments on the application of neural networks in the musical context. The MAX object implemented feedforward, as well as Jordan and Elman type recurrent networks. They applied it to recognition of gestures from a MIDI keyboard, the Radio Drum, the ZETA MIDI guitar, and the Lightning system for continuous spatial control.

In the case of the MIDI keyboard, a neural network allowed to map the keys number, velocity and another control to complex timbre generated by a series of sinusoidal tone generators. This a very typical application where a small number of input parameters is translated into a complex set of parameters for sound synthesis. Lee et al. note that the mapping could have been done by traditional translation functions, but a neural network makes the work less time consuming for humans as it can automate the search for the mapping function by interpolation of input points and generalization.

In the application on the Radio Drum (an array of antennas reporting the distance of transmitters attached to two sticks), a neural network was first used to map a highly nonlinear response of the system to a linear 3D space correctly representing the coordinates of the sticks. Later, the 3D space was bypassed as the data from the system could be directly mapped to a gestural space, further reducing errors.

The output of the ZETA MIDI guitar was first analyzed by traditional means to produce data about the tonal center, note density, note variance, duration, and melodic contour. A neural network was trained to recognize patterns in these features, and use this information to synchronize presentation of images.

The Lightning system featured two infrared transmitters and a receiver, providing spatial coordinates of the transmitters. The gestures produced using the system were analyzed for curvature, symmetry and endpoint distance measures. A neural network was successfully trained on this data to recognize gestures, which, according to the suggestion of Lee et al., could be used for interpretation of conducting gestures.

3.2.2 Le Groux: A Neural Network Principal Component Synthesizer

A similar approach to that of Lee et al., in regard to the MIDI keyboard was developed by Le Groux (2002) (under supervision of Wessel himself), albeit with one important difference: the neural network does not operate on the data extracted directly from the gesture (or MIDI control information), but instead on the features extracted by analyzing the sound produced by an acoustical instrument. The training procedure goes as follows: a recording of a performance on an acoustical instrument is subjected to additive harmonic analysis and then further processed to produce what Le Groux calls *perceptual controllers*: information about pitch, loudness, brightness, etc. These were “chosen in order to correspond to human perceptive criterions and are relevant for musical applications.” (Le Groux, 2002, p.18) The neural network is then trained with this control data as training input, with the corresponding raw harmonic analysis data as teaching output (i.e. desired output). The output could then be used to drive additive synthesis in order to resynthesis the original sound.

Why is this useful? According to Le Groux, the relationship between the abstraction by perceptual controllers on one hand, and the detailed sound data in terms of harmonic spectrum on the other hand, conveys the performer's musical style and identity. Because the neural network learns the internal relationships within the sound data, it is possible for example to play a different melody than the one used at training by supplying a different time series of perceptual controllers (for example using a MIDI control device), while preserving the identity of the original performance. Le Groux describes an experimental example in which an original melody played by Coltrain on the saxophone was changed while supposedly preserving Coltrain's style. Moreover, since the system takes abstract perceptual controllers as input, instead of raw controller data, it is possible to use any controller, given that a mapping from its control outputs to perceptual controllers is done. It is possible, for example, to play the sound of the saxophone with a flute-style controller.

I see several issues in understanding Le Groux's intentions. Firstly, we could ask whether the spectral information can really convey a performer's style. Le Groux does not give any justification for that. Furthermore, he suggests that this system “provides a musician the opportunity to cultivate his/her style” (Le Groux, 2002, p.7), while “the often standardized nature of the available tools limits the musician’s expressive control.” (Le Groux, 2002, p.7) In accordance with my exposition of the issues related to music performance in the second chapter, he states:

> This limit is particularly problematic since musicians are appreciated and recognized for their style. During many long hours of training, a unique relationship develops between the musician and his/her acoustic instrument. The physical properties of the instrument combined with the musician’s individual personality, skills, and creativity, result in a specific playing style. This interaction is the foundation for a musician’s identity, essential for listener recognition and enjoyment. (Le Groux, 2002, p.7)

However, does his system really support the development and cultivation of one's own style? As far as it is apparent from his description, even if the system is really successful in what it does (preserving skill, style and identity), then it allows one to play more sloppily and obtain the same musical results as used in the training data. This is supported by Le Groux's suggestion of the system's usability for studio corrections of recorded material. I can see how this can provide a musician with reflection on own errors and catalyze correction of pure gestural technique (getting the pitches right), but it seems that what it really does is to isolate the gestural control - the performer's active engagement - from the style, and prevent the latter's modification, and hence any development. Simply stated: preservation opposes change, and this approach seems to opt for the former.

3.2.3 The Wekinator

Another recent example of artificial neural networks used for gesture-to-sound-synthesis mapping is a recently developed software named the Wekinator (Fiebrink, Truman and Cook, 2009; Fiebrink, 2011). This software offers a wider range of machine learning algorithms (others aside from neural networks), with one important intent: it focuses on the ability of a performer to create a highly personalized mapping between gestures (tracked by any kind of controller that can interface with a computer) and sound synthesis parameters. The emphasis is on an individual performer developing the machine learning scheme entirely on their own, for their own purposes and for a single performance/composition. This is therefore an approach that fits well under the paradigm of "instrument composition" (Murray-Brown et al., 2011) that I have presented in the second chapter. A lot of attention was therefore given to the design of the user interface, which was developed with the help of feedback evaluations by various performers using the software for customized mappings employed in real performances. Fiebrink explains what these conditions imply:

> Among the interactional affordances that were key to making standard supervised learning algorithms usable in this work were their low training time, their capability for building models for the chosen learning concepts using a small number of training examples, their fast running time, and their ability to be "steered" in different directions via users' modifications to the training set. (Fiebrink, 2011, p. 367)

There is no demand for great generalization by the neural networks, since the scope of learning is much smaller than when trying to create a general-purpose gesture recognition system. Moreover, due to great insight that the software offers into the machine learning process, the performer can adjust the patterns to be learned according to the feedback about the success of learning; if a set of patterns can not be learned, it can simply be adjusted. But most importantly, this provides the performer with the information about the consistency of their own actions and self-perception: a neural network can only learn to associate patterns if there is indeed an inherent connection, and distinguish them if there is an inherent distinction. If the performer tried to map what they believed is a prominent feature of their interaction with the instrument, but the feature was not conveyed precisely and clearly enough through the gestures, it would quickly show the neural network's inability to learn. The same holds for the other side of the training: the target features of the sound synthesis to be mapped.

Due to the Wekinator's great flexibility and focus on rapid development of gesture-to-sound mapping, Fiebrink et al. (2009) propose its usage in real-time performance, for on-the-fly machine learning. This is especially supported by the software's ability to integrate playing and machine learning in a tightly unified process:

> What the Wekinator system encourages is a high-level, intuitive approach, where particular mapping “nodes” can be quickly defined via training examples and then the instrument immediately auditioned. Rather than laboriously mode-shifting, the instrument builder now takes part in a playful process of physically interacting with a malleable complex mapping that can be shaped but does not have to be built from the ground up. Furthermore, the surprises that the mapping inevitably generates, while sometimes undesirable, are often inspiring. Being able to save these mappings and revisit them, perhaps modifying them on-the-fly in performance, allows for continuity but also continual evolution. (Fiebrink et al., 2009, p. 5)

The Wekinator used in on-the-fly machine learning in live music performance can embody the three instrument design principles proposed by Murray-Brown et al. (2011): 1. The instrument is designed for an individual performer, for a single performance - obviously, since the performance is an improvisation, but moreover the interactivity and the instrument itself are being designed during the performance. 2. The complexity of interactivity gradually increases as the performance develops. 3. Potentially, the element of surprise on the audience is absolute with every instance of the performance. But what is most valuable, in my opinion, is that the decision to begin with an almost empty interaction scheme is a *real* constraint imposed on the performer, and the development of the interactivity comes from the exploration and gradual expansion of real limitations, as an answer to them. I suggest that this plays an important role also on the ability of the audience to understand the musical drama on stage.

4. My Software and Musical Performance

4.1 Artificial Neural Network Extensions for SuperCollider

As a prerequisite to my experimentation with the application of artificial neural networks in a computer instrument, I developed several software components that allow more efficient use of feedforward neural networks in the SuperCollider environment than was previously possible. The components are a plug-in system for the SuperCollider language, a plug-in for the language, utilizing this system, and several UGens for the SuperCollider synthesis server. Both the language plug-in and the UGens make use of the Fast Artificial Neural Network library (FANN). To my knowledge, this is the first implementation of neural network computation capabilities for SuperCollider that compiles into native machine code, and is thus computationally much more efficient than similar capabilities implemented in the SuperCollider language itself.

The SuperCollider plugin interface built into the latest stable version of SuperCollider at the time of this writing is available at:
https://github.com/jleben/supercollider/tree/topic/lang-plugins-3.5.2

The neural network extensions are available at:
https://github.com/jleben/supercollider-ann

Although the SuperCollider synthesis server has long supported plug-ins, the language has never had a plug-in interface, and it is not available at the latest version of the software.

Since the primary intention was to develop a natively compilable neural network extension for the language, this was a good incentive to enrich the language with a plug-in interface. The neural network capabilities could have been compiled directly into the language, but the ability to implement them in the form of a plug-in increased the maintainability of the code, while the effort to develop a general plug-in system naturally works towards the benefit of a larger community.

The language plugin implements a SuperCollider language class (named Fann) that represents a feedforward neural network. It is essentially an interface to a subset of the FANN library API. It allows to create feedforward networks of arbitrary amount of layers with arbitrary amount of neurons, with each layer fully connected to the subsequent layer. At the time of this writing, no shortcut connections are yet possible. Different activation functions can be specified for the hidden layers, and for the output layer, and this can be any activation function supported by FANN. The network is initialized with random weights, and can be reset into a new random state at any time. It can be trained using the FANN's default training algorithm, which is Rprop, or resilient backpropagation, an enhancement of the traditional backpropagation-of-error algorithm. The training can be performed epoch-by-epoch, or automatically until a desirably small error value is reached. The class leverages FANN's capabilities to save the complete state of the neural network into a file, and load a saved networked. This capability is also used as a way to transfer a trained neural network to the SuperCollider server, where it can be loaded by a UGen.

The server plug-in contains additional infrastructure, aside from the UGens. This infrastructure allows for safe loading of neural networks saved to files, without interference with the audio computation. Loading data from files is not a real-time safe operation, as it never has a strictly determinable time of completion, and hence must not be performed on the same computing thread on which the audio processing runs, or it could take too long time and interrupt the audio. Therefore, upon receiving a specific synthesis server message from the language, the plug-in loads a neural network from file into one of the numbered slots in a non real-time thread, and takes care that a UGen instantiated on the real-time thread can access the neural network in a thread-safe manner.

Currently, there are two useful UGens implemented; their corresponding language classes are named AnnBasic and AnnTime. Both take a slot index as the first parameter, which defines one of the slots at which they will try to access a loaded neural networks (if any exists there).

The AnnBasic UGen operates at control rate. It takes an array of control rate inputs as the second parameter, of which each will be mapped to one neural network input. For the fact that the synthesis server architecture does not allow dynamic creation of UGen inputs and outputs after they are instantiated, the number of outputs is specified as the third parameter, and both this number, as well as the size of the input array at the second parameter must match the structure of the loaded neural network. The AnnBasic UGen thus simply maps its inputs and outputs to inputs and outputs of a neural network.

The AnnTime UGen, however, is intended for processing of a time-series. It can operate both in control and in audio rate. As the second parameter it takes a window size (W, in continuation)*,* and as the third parameter a single data input. It also has a single output. Internally, it contains a buffer of the size W where it stores W amount of past input values incoming on the data input. At every computational step (an audio sample at audio rate, or once per each control rate step) it presents the buffered input values as the input vector to the neural network, and after processing the network, forwards the network's output through its own output. For this UGen, the neural network denoted by the first parameter must match the specified window size in number of inputs, and must have one single output.

4.2 The Instrument and the Performance

I will describe the computer instrument employing the above presented extensions for the SuperCollider environment that I used in my live improvisation performance at the Institue of Sonology on 18 April 2012.

The fundamental idea was to explore the way that neural networks - trained to analyze a particular simple input time series, and map it into into a more complex output time series - would modify the latter as the input was modified. Another starting point was that several such neural networks would be used to contributively shape various aspects of a common sound body, instead of several preceptively separated sound entities. Moreover, the definition of which neural network contributed to which aspect of sound, and in what amount, would be determined during the performance - starting with a very simple scheme, and then gradually increasing the complexity of mapping, applying a neural network to several aspects of sound, or several neural networks to the same aspect of sound.

The neural network processing was performed at control rate, so the outputs of several neural networks were routed into another process as control data for sound synthesis. In this configuration, the sound synthesis served as a kind of sonification method for the output of the neural networks, and I was primarily interested in musical features conveyed on the control rate level. The sound synthesis was chosen to be traditional frequency modulation (FM). The reason for this is that the FM technique has a well defined paradigm for construction of an FM graph consisting of individual sinusoidal oscillators, whereby one or more oscillators are summed together to modulate the frequency of another oscillator. This lent itself well to the idea of mapping the neural network outputs to different aspects of sound: this was accomplished by applying the neural network outputs to carrier frequencies or modulation indexes of various parts of the FM graph. New mappings could be established or old one broken during the performance.

The most interesting part, however, was happening in relation to the neural networks. They were trained to map one period of a sinusoid as a time series to a different, arbitrarily complex time series. When a sinusoid of the same frequency as used in training would run as an input into such a neural network, the situation was equivalent to the *teaching input* (see chapter 3.1.4) being stored in a table, and the network input playing a role of a running phase for table lookup. In this case, the whole system can be seen as analogous to a wavetable oscillator. However, when specific kinds of slight modifications to the input signals were introduced, the output would be nonlinearily modified. Now, one can ask immediately: why not simply perform a table lookup with a nonlinear phase function? The reason is that this would require one to *produce* the nonlinearity in the phase function. On the other hand, a simple *offset* (addition of constant value, or DC) of the input signal into the neural network would already produce nonlinear modifications of the output. And the amount of offset thus becomes a powerful parameter to be exposed as a physical control.

In other words, the above paragraph simply states that the whole system observed with the sinusoid offset parameter as an input, and the neural network as an output, is nonlinear. So the question arises, why this particular nonlinear system, and not any other? I found this system particularly interesting. I will describe several interesting behaviors that I observed. When I speak of change in the input signal, I refer to change in relation to the signal used at training:

- When changing the frequency of the input sinusoid the wavetable oscillator analogy would still hold: the output would mostly just change its fundamental frequency proportionally, but preserve the shape (i.e. relative amplitudes and phases of harmonics).
- When decreasing the amplitude of the input, the output would attenuate accordingly. However, when increasing the amplitude, the features of the output would seem to amplify, but preserving the total energy. The slopes would get steeper without affecting the peaks. This was due to the fact that such neuron activation functions were used which have a bounded range, and can not possibly output values outside that range.
- When adding offset to the input, the various features of the output would seem to shift in time in different directions, as if we were changing phases of groups of partials.
- When adding sinusoids of different frequencies on the input, the output would reflect corresponding different periods, akin to combining several instances of the original signal at different frequencies, but in a manner more complex than summing.

The final system used in the performance featured four neural networks trained in the manner described above. Each neural network had two oscillators plus an offset value summed together as the input. During the performance, the oscillators were being tuned in the range between 0 and 20 Hz. On the sound synthesis side, there were four predefined FM graphs. The outputs of each neural network could be applied to either the frequency or the modulation index control at various points in any of the FM graphs, in various amounts.

A lot of parameters were controllable using a physical controller (either faders or rotary knobs). A mapping between a neural network output and a point in an FM graph was established by means of pressing two physical buttons (one for source and one for target selection). The amplitudes of the oscillators and the offset values were the most used controls in the performance; they provided a kind of gestural control, so they were the only ones controllable with faders. The carrier frequencies and amplitudes of the final oscillators of the FM graphs (providing audio output) were controlled using rotary knobs. And finally the amount to which the neural networks would affect the mapped parameters of the FM synthesis were controlled using rotary knobs. The frequencies of the oscillators had preset values, and not changed often during performance, so they were only controllable in software.

The neural networks were trained with newly generated patterns as teaching inputs just before the performance, each with a distinct pattern (or table, to repeat the wavetable oscillator analogy). At the start of the performance, there was no mapping between a neural network and the FM synthesis established. The amplitude of only one oscillator per neural network was set to 1, the other oscillator and the offset to 0. This ensured that when a neural network was mapped and activated, it would first produce the exact pattern used in training.

The initial patterns provided a point of departure for exploration. As I had not trained with these exact patterns before, it was unpredictable how exactly they would change according to the input controls, although I knew the general principles listed above. It was intriguing, however, that the change was dependent on the content of the patterns themselves. It was the inherent features of the patterns that the networks learned, and that was what they operated on. As the deviations were gradually introduced during the course of the performance, I would learn increasingly more about the patterns. The performance was therefore a play between the moments of stepping into unknown, learning more about the response of the system, integrating that knowledge into the total sound world, and repeating the same again.

Conclusion

The introduction of the computer into the music performance has brought a considerably different dynamics onto the stage than those seen at a performance with acoustical instruments. With the computer taking a lot of autonomy in contribution to the music, the notion of the performer's skill and virtuosity has suddenly lost ground. Questions about what is a performance at all have arisen.

It appears that the discourse about this issues has been drastically intensifying over the last couple of decades, and it is certainly extremely active at present. Throughout all this time, the question of virtuosity has been in the center of attention. While it seems that we can not do away with the term, the term itself may be changing meaning.

It is important to realize that the issue of virtuosity in relation to the computer is twofold: on one hand it is the issue of distancing of musical output from the control of the performer; on the other hand it is the issue of rapid development and invention of novel instruments, so that a paradigm of use does not have time to form. In this view, the notion of virtuosity does not have space while it is considered as something static, accumulating over the course of one's life or history as a sediment. When the only constant is reinvention of the instrument, we are in a constant state of learning. The skill supporting virtuosity must become the skill of learning.

In my conclusion, an approach to music performance that takes the development and the exploration of a new instrument as the musical topic in itself may pull the discourse about virtuosity in a favorable direction. Aside from providing a clear enough meta-paradigm, that allows for actual development of a new kind of virtuosity, it also has the potential to engage the audience in a shift towards a better appreciation of the music.

I can identify my employment of artificial neural networks for the purpose of musical improvisation as a small attempt within this paradigm. In its core it embodies an exploratory interactivity scheme, and invites to exploration on the basis of constraint, rather than abundance.

However, I see a lot of room for improvement, especially in regard to two topics:

1. Making the neural network implementation efficient enough to run a satisfying amount of neural networks at audio rate. The purpose of this would be to let the neural network have more direct control on the audio output, and thus convey more of its operation through sound.

2. Improving the physical control: decreasing the dimensionality of human control inputs or/and a different kind of physical controller could yield a great improvement in interactivity.

References

Chadabe, J. (1984). Interactive Composing: An Overview. *Computer Music Journal*, 8(1), 22.

Chadabe, J. (2002). The limitations of mapping as a structural descriptive in electronic instruments. In *Proceedings of the 2002 conference on New interfaces for musical expression* (pp. 1-5). Singapore: National University of Singapore.

Fiebrink, R. (2011). *Real-time Human Interaction with Supervised Learning Algorithms for Music Composition and Performance* (Doctoral dissertation, Princeton University, 2011). Retrieved March 4, 2012, from http://www.cs.princeton.edu/~fiebrink/drop/finalthesis/RebeccaFiebrinkThesisPQ.pdf

Fiebrink, R., Trueman, D., & Cook, P. R.. (2009) A meta-instrument for interactive, on-the-fly machine learning. In *Proceedings of the 2009 International Conference on New Interfaces for Musical Expression.* Pittsburgh.

Gurevich, M., Stapleton, P., & Bennett, P. (2009). Designing for Style in New Musical Interactions. In *Proceedings of the 2009 Conference on New Interfaces for Musical Expression* (pp. 213-217).

Gurevich, M., Stapleton, P., & Marquez-borbon, A. (2010). Style and Constraint in Electronic Musical Instruments. In *Proceedings of the 2010 Conference on New Interfaces for Musical Expression* (pp. 106-111). Sydney, Australia.

Hallas, M., & Dorffner, G. (1998). *A Comparative Study on Feedforward and Recurrent Neural Networks in Time Series Prediction Using Gradient Descent Learning*. Retrieved March 3, 2012, from http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.50.6673&rep=rep1&type=pdf

Kriesel, D. (2007). *A Brief Introduction to Neural Networks*. Retrieved May 12, 2012, from http://www.dkriesel.com

Le Groux (2002): *A Neural Network Principal Component Synthesizer for Expressive Control of Musical Sounds* (Dissertation, Ircam Pedagogy Department, 2002). Retreived from: Retrieved March 15, 2011, from http://www.atiam.ircam.fr/Archives/Stages0102/SylvainLegroux.pdf

Lee, M., Freed, A., & Wessel, D. (1991). Real-time neural network processing of gestural and acoustic signals. In *Proceedings of the International Computer Music Conference* (pp. 277-277). International Computer Music Association.

Marquez-Borbon, A., Gurevich, M., Fyans, A. C., & Stapleton, P. (2011). Designing Digital Musical Interactions in Experimental Contexts. In *Proceedings of the 2011 International Conference on New Interfaces for Musical Expression* (pp. 373-376). Oslo: ???

Murray-Browne, T., Mainstone, D., Bryan-Kinns, N., & Plumbley, M. D. (2011). The medium is the message : Composing instruments and performing mappings. In *Proceedings of the 2011 International Conference on New Interfaces for Musical Expression* (pp. 56-59).

Schloss, W. A., & Jaffe, D. A. (1993). Intelligent musical instruments: The future of musical performance or the demise of the performer? *Journal of New Music Research*, 22(3), 1-9. Retrieved from http://people.finearts.uvic.ca/~aschloss/Articles/INTERFACEarticle.html

Schloss, W. A. (2003). Using Contemporary Technology in Live Performance: The Dilemma of the Performer. *Journal of New Music Research*, 32(3), 239-242.

Wessel, D., & Wright, M. (2002). Problems and Prospects for Intimate Musical Control of Computers. *Computer Music Journal*, 26(3), 11-22.

Wessel, D. (2006). An Enactive Approach to Computer Music Performance. *Le Feedback dans la Creation Musical*, 93-98. Lyon, France: Studio Gramme.