

# J.S. Bach vs. HAL9000: Strijd tussen de Componisten

De potentie van neurale netwerken voor het componeren  
van creatieve melodieën

Student: Pieter Pierrot  
Studentnummer: 3993469

Eerste beoordelaar: J.M. Broersen  
Tweede Beoordelaar: T.B. Klos  
ECTS: 7,5  
Datum: Januari 2017

Universiteit Utrecht  
Departement Geesteswetenschappen  
Kunstmatige Intelligentie



**Universiteit Utrecht**

## Samenvatting

In deze tekst onderzoek ik of een computerprogramma in staat is melodieën te componeren die door mensen als creatief gezien kunnen worden. Ik zal een suggestie doen voor een uitbreiding van het programma van Mozer en Soukup uit 1991, CONCERT. CONCERT onderscheidt zich van andere programma's door middel van een psychologisch gegronde representatie van toonhoogte, die perceptuele gelijkenissen tussen verschillende noten waarborgt. Echter, de melodieën die CONCERT componeert missen globale structuur en thematische coherentie. Ik zal beargumenten waarom het systeem van Mozer en Soukup niet in staat is tot het componeren van creatieve melodieën en ik zal een modificatie van het systeem beschrijven die voor dit doeleinde potentie heeft.

**Keywords:** Kunstmatige Intelligentie, Algoritmische Compositie, Computationale Creativiteit, RNN's

*Time's existence cannot be found between the tick and the tock of a clock. It is the language of life and, as such, is most powerfully felt in the context of human experience.*

— **Robert Lanza, in *Biocentrism***

# Inhoudsopgave

<b>1</b>	<b>Introductie</b>	<b>3</b>
1.1	Kunstmatige Intelligentie en creativiteit . . . . .	3
1.2	Kunstmatige Intelligentie en compositie . . . . .	4
1.3	Huidig onderzoek . . . . .	6
<b>2</b>	<b>Mozer en Soukup's CONCERT</b>	<b>8</b>
2.1	Recurrent Neural Networks (RNN's) . . . . .	8
2.2	CONCERT's PHCCCF representatie . . . . .	10
2.3	CONCERT's sterke en zwakke kanten . . . . .	12
<b>3</b>	<b>Componeren met structuur en thema's</b>	<b>14</b>
3.1	Structuur en thema's in muziek . . . . .	14
3.2	Structuur en thema's in RNN's . . . . .	15
3.3	Structuur en thema's in CONCERT . . . . .	16
<b>4</b>	<b>CONCERT 2.0: mijn aanpassing</b>	<b>18</b>
4.1	Complexiteit van programma's . . . . .	18
4.2	Notenclusters van verschillende groottes . . . . .	18
4.3	Parallele RNN's . . . . .	20
<b>5</b>	<b>Potentie en Discussie</b>	<b>22</b>
5.1	CONCERT 2.0 in de praktijk . . . . .	22
5.2	Filosofische overwegingen voor vervolgonderzoek . . . . .	23
<b>6</b>	<b>Bibliografie</b>	<b>26</b>

# 1 Introductie

## 1.1 Kunstmatige Intelligentie en creativiteit

De gedachte dat de potentie van het vak dat Kunstmatige Intelligentie (KI) heet onbegrensd is, is niet zeldzaam. Zowel onder wetenschappers als in de publieke opinie bestaat de angst dat de wetenschap computers zal kunnen smeden tot supermensen, van wie de motieven en doelen niet te doorgronden zijn. De grenzen van deze smeedkunst zijn, zo is de gedachte, van technologische (en daarmee tijdelijke) aard, en niet van een conceptuele. Ze worden bepaald door de denkkraft van de computers, en de grootte van de denkkraft groeit snel. De beroemde wet van Moore garandeert nog steeds de verdubbeling van denkkraft elke twee jaar en staat daarmee als vlag aan de horizon voor wetenschap en technologie-bedrijven. Buiten het feit dat de wet van Moore de dichtheid van transistoren beschrijft, en in bredere contexten strict gezien alleen als metafoor gebruikt kan worden (Moore, 1995), denk ik dat er op de weg vele hordes staan die overwonnen zullen moeten worden.

Deze tekst zal gaan over een van die hordes: creativiteit. Wanneer ik denk aan het debat rond het gevaar van KI, dan denk ik aan slimme computersystemen waarvan de agenda buiten de controle van de mensen valt. Deze systemen, die in staat zijn autonome beslissingen te nemen, zouden hun superieure denkkraft dan kunnen inzetten voor doeleinden die gevaren zouden vormen voor mensen. Het maken van zulke autonome beslissingen vereist echter een creatief aspect in de zin dat er nieuwe dingen bedacht zullen moeten worden. In deze onderschatte of vergeten creatieve stap ligt mijn interesse.

Creativiteit is een illuster woord in de KI, waarvan de berekenbaarheid nog niet een serieus veld van onderzoek is. Creativiteit staat immers hoog op de rangorde van onbegrip omtrent het menselijke cognitieve begrippenapparaat, en in wetenschappelijke kringen is het niet onomstreden. Over een heldere definitie is men het niet eens (Boden, 1994).

Het is niet mijn bedoeling me in een discussie omtrent de inhoud van die definitie te mengen; die discussie laat ik in de handen van filosofen en neurowetenschappers. Het gebruik van het woord creativiteit in dit onderzoek is met een bredere context in gedachten. Dat een goede definitie niet voorhanden is, betekent namelijk niet dat het woord creativiteit voor mij geen bruikbare inhoud heeft. Ik wil het woord gebruiken met een brede basis, waarbij ik een soort huis-tuin-en-keuken definitie wil hanteren: mensen

zijn creatief wanneer ze kunstwerken maken, maar ook wanneer ze zich inleven in een boek, voetballen, knutselen, huilen bij films, bordspellen spelen of naar een circusvoorstelling kijken. Immers, bij elke van deze activiteiten begeven mensen zich op nieuw terrein en integreren ze nieuwe informatie om tot nieuwe ideeën, gedachtes of gevoelens te komen. Niemand weet wat creativiteit is, men kan het zelfs ernstig oneens zijn, maar iedereen heeft er wel degelijk een conceptie van, zij het nog zo vaag: bakstenen zijn niet creatief, Rembrandt van Rijn was dit wel. We zullen zien dat zo'n onafgebakend onderscheid geen probleem vormt voor het trekken van interessante conclusies.

Omdat zo'n gebruik van het woord creativiteit niet is ontwikkeld in de computationele kant van de KI, zal het onderzoeken ervan volgens mij ook een menselijke kant moeten krijgen. Ik zal dan ook geen model of objectief criterium gebruiken dat me moet vertellen of een algoritme of een programma creatief is of niet. Zo'n aanpak zou mijn menselijke interpretatie van creativiteit verloochenen. Ik zal de mens zelf als maatstaf gebruiken. Als creativiteit een woord is dat door mensen wordt gebruikt en uitgesproken wordt, moeten zij bepalen of zo'n uiting van een programma al dan niet creatief is. Ik denk dat dit op dit moment geen problemen gaat opleveren, omdat het veld waarin ik mijn onderzoek doe nog niet ver ontwikkeld is. Ik verwacht dat vele pogingen tot het schrijven van creatieve melodieën niet succesvol zijn, en geen stof zullen zijn voor debatten tussen luisteraars. Op het moment dat zulke debatten wel zullen plaatsvinden, zal het wellicht noodzakelijk zijn dit standpunt te herzien.

Ik wil geen verwarring veroorzaken over het onderwerp van onderzoek. Dit zal een onderzoek worden naar de potentie van KI als computationeel veld voor de implementatie van een concept dat tot zover slechts aan mensen werd toegeschreven. Het veld van computationele creativiteit mag dan nog in zijn kinderschoenen staan, er zijn al eerste voorzichtige fluisteringen hoorbaar over creativiteit in programma's. In 2016 versloeg AlphaGo, een programma van Google, de wereldkampioen van het Aziatische bordspel Go. Churchland en Sejnowski (2016), beschreven het als volgt: *"AlphaGo displayed human-like creativity, demonstrating the power of reinforcement learning coupled with deep learning"*.

## 1.2 Kunstmatige Intelligentie en compositie

Een goede plek om te zoeken naar creativiteit in kunstmatige intelligentie is in mijn ogen muziek die geschreven is door een programma. Muziek is groot

onder de kunstvormen die we als mensen hebben voortgebracht en behelst veel van onze tijd, geld en interesses. Ook wordt het woord creativiteit veel in verband gebracht met mensen die muziek maken (denk aan een scala van menselijke beroepen, waaronder componisten, bandleiden, zangers, en violisten). De keuze had ook kunnen vallen op het uitvoeren van muziek, hier schuilt immers ook een hoop creatieve potentie, maar dit is voor een groot deel een probleem van de robotica en dit is niet waar mijn interesse ligt. Compositie door computerprogramma's, dat bekend staat als algoritmische compositie, is een zeer geschikt onderwerp voor mijn onderzoek.

Er is, zoals je zou verwachten, een hoop onderzoek gedaan in het veld van algoritmische compositie<sup>1</sup>. Papadopoulos en Wiggins schrijven in hun survey (1999) de eerste poging van algoritmische compositie op basis van een computationeel model toe aan Hiller en Isaacson in 1959. Sindsdien zijn er een hoop verschillende methoden gebruikt, met elk hun sterke en zwakke kanten (Papadopoulos en Wiggins, 1999). Zo zijn er de Knowledge Base Systems (KBS), die gebruik maken van een expliciete set regels of beperkingen (de Knowledge Base) (e.g. CHORAL, 1990), methoden die muziek bekijken als een grammatica, waarbij verschillende delen van die grammatica moeten worden gecombineerd om een muzikaal stuk te vormen (e.g. EMI, 1984), en de evolutionaire aanpak, die genetische of evolutionaire algoritmes gebruikt om een steeds betere set van composities te krijgen (e.g. Papadopoulos en Wiggins, 1998).

Ik heb gekozen me te richten op het gebruik van neurale netwerken (of: de connectionistische aanpak) om de simpele reden dat de neurale netwerken de werking van het menselijk brein imiteren. Omdat ik creativiteit wil onderzoeken zoals gebruikt en gezien door mensen, is het logisch een methodische aanpak te bestuderen die dicht bij het menselijke cognitieve systeem blijft. De kans op composities die de Turing-test van creativiteit doorstaan is immers het grootst wanneer we het compositieproces modelleren naar die van de mens.

CONCERT (**C**onnectionist **C**omposer of **E**rudite **T**unes), in 1991 opgesteld door Michael Mozer en Todd Soukup, is zo'n connectionistisch programma. CONCERT maakt gebruik van een RNN, een Recurrent Neural Network, om onder meer melodieën te schrijven in de stijl van Johann Sebastian Bach. Om een aantal redenen waarover ik hierna zal uitweiden, heb

---

<sup>1</sup>Niemand minder dan W.A. Mozart zelf publiceerde in 1792 al een dubbelspel waarin muziek geschreven werd aan de hand van de uitkomst van een reeks dobbelsteenwerpen.

ik gekozen me voor mijn onderzoek te focussen op dit programma.

### 1.3 Huidig onderzoek

Wat mij snel duidelijk werd toen ik had besloten algoritmische compositie te onderzoeken, was dat ik me wilde gaan focussen op oudere klassieke muziek, zoals de muziek van Bach, Mozart of Haydn, in plaats van op hedendaagse muziek zoals pop- of rockmuziek. Hier is een duidelijke reden voor: in hedendaagse muziek ligt de nadruk minder op de compositie (noten, ritmes, thema's etc.) en meer op de sound (hoe *klinkt* het?) van de muziek. Deze sound-gebaseerde aanpak is minder geschikt voor algoritmische compositie, omdat het schrijven van een melodie lang niet genoeg is voor het verkrijgen van een interessant (creatief) stuk hedendaagse muziek. Een nummer van Radiohead of Adele gaat door een uitgebreid productieproces (achtergrondzang moet ietsje harder, gitaar klinkt 'te ruw' etc.) voordat het goed genoeg wordt geacht voor een album, terwijl de klassieke muziek van Bach direct vanaf de bladmuziek werd uitgevoerd (er bestond natuurlijk nog geen manier om muziek op te nemen)<sup>2</sup>. Zijn muziek draait om de compositie, waardoor het geschikter is voor computationele doeleinden dan hedendaagse muziek.

CONCERT is een programma dat gebruikt is om te leren melodieën te schrijven in de stijl van Bach. Dit maakt het een uitstekend programma voor de problemen die ik in mijn onderzoek wil adresseren en het is dan ook een van de belangrijkste redenen voor mijn keuze voor de focus op CONCERT.

Een andere reden voor deze keuze is een eigenschap van het programma zelf: wat CONCERT onderscheidt van andere programma's die melodieën componeren, is de bijzondere representatie van toonhoogte, waarvan ik de details in hoofdstuk 2 zal bespreken. Toonhoogte is, vanzelfsprekend genoeg, van groot belang voor het componeren van melodieën: zonder verschillende tonen van verschillende hoogten zou een melodie niet kunnen bestaan. In overeenstemming hiermee, de keuze voor hoe verschillende tonen zich tot elkaar verhouden in een programma is bepalend voor het succes van de melodie als creatieve creatie.

---

<sup>2</sup>Toen Bach in Leipzig woonde, schreef hij voor elke zondag een cantata om de dienst te supplementeren. Dit betekent dat Bach en de muzikanten een week hadden om de muziek te componeren, op te schrijven (voor elke muzikant een partituur), in te studeren en uit te voeren. Bach had daarnaast ook nog andere muzikale klussen die gedaan moesten worden.



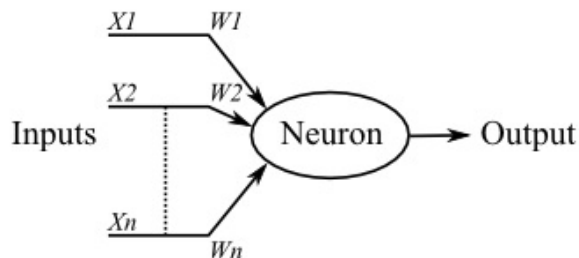
Ondanks alle eigenschappen van CONCERT die het een geschikte kandidaat voor mijn onderzoek maken, zijn de composities naar mijn mening niet creatief. Niet alleen is het duidelijk dat de melodieën zijn geschreven door een computerprogramma, ook is het duidelijk dat dit programma zijn oorsprong heeft in een wetenschappelijk veld dat nog in zijn kinderschoenen staat. Omdat ik geïntrigeerd ben door de architectuur van CONCERT, wil ik een suggestie doen voor een uitbreiding die meer potentie heeft voor het componeren van creatieve melodieën. Om duidelijk te maken waarom ik denk dat deze uitbreiding potentie heeft, zal ik eerst in detail ingaan op de eigenschappen van CONCERT, waaronder de architectuur van het programma, de speciale representatie van toonhoogte en de sterke en zwakke kanten. Daarna zal ik beschrijven waarom ik de melodieën die door CONCERT geschreven worden niet als creatief kan bestempelen. Als we deze zaken op orde hebben, hebben we genoeg kennis om te bestuderen of mijn uitbreiding wel creatieve melodieën kan schrijven.

## 2 Mozer en Soukup's CONCERT

### 2.1 Recurrent Neural Networks (RNN's)

Om te begrijpen hoe CONCERT het probleem van compositie benadert, moeten we eerst kijken naar de algemene technieken die gebruikt worden. Daarom zal ik een korte introductie geven in neurale netwerken. Immers, CONCERT is zo'n neuraal netwerk. Deze introductie zal de wiskundige details buiten beschouwing laten, en alleen een globaal overzicht geven van hoe neurale netwerken problemen oplossen.

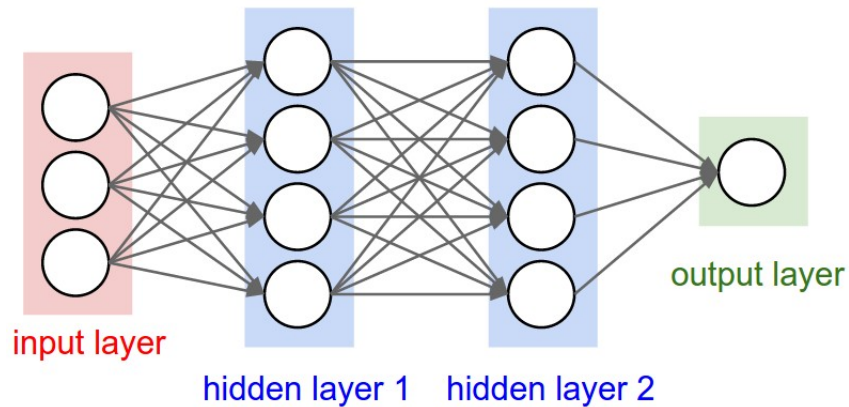
Een neuraal netwerk (ANN, van Artificial Neural Network) is een verzameling van verbonden knopen, waarbij elke knoop een neuron uit een menselijk brein voorstelt. Dit houdt in dat elke knoop kan 'vuren', en daarmee zijn bijdrage levert aan de verwerking van informatie. Of een knoop vuurt, hangt af van zijn verbindingen met andere knopen, de



**Figuur 1:** Een knoop uit een neuraal netwerk, met  $X_1 \dots X_n$  de invoer van het neuron en  $W_1 \dots W_n$  de gewichten van de verbindingen

sterkte van die verbindingen (gewichten) en het al dan niet vuren van de knopen waarmee de knoop in verbinding staat (Figuur 1). Er ontstaat dan een ANN door een groot aantal knopen met elkaar te verbinden. Een ANN is normaliter opgebouwd uit verschillende lagen knopen, waarbij elke laag verantwoordelijk is voor een specifieke functie van de computatie: de invoerlaag, waar de testdata in het netwerk worden gevoerd, een of meerdere verborgen lagen, die de data transformeren tot een bruikbare vorm voor de laatste laag, de uitvoerlaag, die een uiteindelijke representatie van de uitvoer draagt. Zie figuur 2 voor een simpele voorstelling van een ANN.

De probleemoplossende kracht van een ANN zit hem in de matrix van gewichten tussen de knopen. Voor elk probleem bestaat een andere matrix, en de matrix wordt door het netwerk geleerd door het te trainen op probleemspecifieke testdata. Dit trainen gebeurt door de testdata stapsgewijs



**Figuur 2:** Een simpel neurale netwerk met twee verborgen lagen

aan het netwerk te voeren, en na elke stap de gewichtenmatrix te updaten met het algoritme dat Backpropagation heet. Dit mechanisme van een lerend netwerk is direct afgeleid van ons wetenschappelijk begrip van hoe het menselijk brein werkt.

Ten opzichte van andere computationele technieken hebben ANN's een aantal goede eigenschappen. Ze hebben geen expliciete kennis nodig, zoals KBS's die ik heb genoemd in het openingshoofdstuk, omdat ze alle informatie extraheren uit de invoerdata. Daarnaast zijn ze goed in staat patronen en regelmatigheden in deze invoerdata te herkennen. Vanwege deze eigenschappen is het gebruik van ANN's een populaire aanpak in algoritmische compositie (Papadopoulos en Wiggins, 1999).

Een veelgebruikte specificatie van de ANN is de RNN, wat staat voor Recurrent Neural Network. RNN's zijn net als gewone ANN's, met de toevoeging van feedbackloops tussen knopen in het netwerk. Dit houdt in dat uitvoer van knopen in een bepaalde laag worden gebruikt als invoer voor andere knopen in dezelfde of eerdere lagen (recurrence). Een RNN kan volledig recurrent zijn, wat inhoudt dat elke knoop in feedbackverbinding staat met elke andere knoop. CONCERT maakt gebruik van een niet-volledige recurrente RNN.

De feedbackloops kunnen gebruikt worden als een intern geheugen, waardoor in het programma temporele informatie kan worden opgeslagen. In algoritmische compositie wordt zulke informatieopslag gebruikt, bijvoorbeeld om temporele cues te onthouden over hoe ver het programma is in zijn compositie of welke noot er meerdere noten geleden is geschreven.

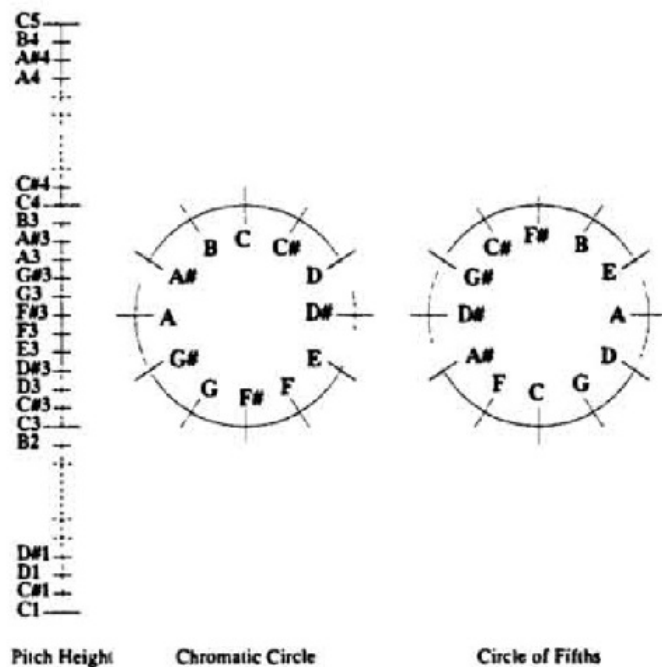
## 2.2 CONCERT's PHCCCF representatie

Nu we een idee hebben van hoe een ANN gebruikt kan worden om computationele problemen op te lossen, wordt het tijd om te gaan kijken naar de werking van het specifieke ANN CONCERT. Zoals gezegd is een belangrijke reden voor mijn keuze voor CONCERT als onderwerp van dit onderzoek de bijzondere representatie van toonhoogte. Deze representatie vindt zijn oorsprong in het psychologisch-musicologisch onderzoek van Roger Shepard uit 1982. Shepard stelde naar aanleiding van dit onderzoek een generalisatietheorie op over hoe mensen de gelijkensissen tussen noten van verschillende hoogtes (intervallen) waarnemen. Hij stelt in zijn theorie dat deze perceptuele gelijkensissen passen in een vijf-dimensionale vectorruimte: een dimensie voor toonhoogte, twee dimensies (x- en y-coördinaten) voor de chromatische cirkel en twee voor de kwintencirkel<sup>3</sup> (Figuur 3). Twee verschillende noten verschillen van elkaar in minstens een van deze dimensies, en de manier waarop de noten van elkaar verschillen in elke dimensie bepaalt volgens Shepard het verschil in waarneming van luisteraars. Zo worden bijvoorbeeld twee noten die ver uit elkaar liggen op de dimensie van toonhoogte, maar slechts een stap uit elkaar liggen op de kwintencirkel toch waargenomen als noten met grote gelijkensissen.

De auteurs van CONCERT hebben dit de PHCCCF-representatie genoemd, naar de onderdelen waaruit het is opgebouwd (**P**itch **H**eight, **C**hromatic **C**ircle, **C**ircle of **F**ifths). In CONCERT is een directe vertaling van deze theorie gebruikt, op maat gemaakt voor het gebruik in een neurale netwerk. Het is niet moeilijk voor te stellen waarom deze representatie interessant is voor mijn onderzoek: het ademt de menselijke ervaring die het uitgangspunt is voor mijn idee van creativiteit. Ondanks dat er onderzoekers zijn die beweren dat de keuze voor een bepaalde programma-interne representatie geen invloed heeft op de werking van het neurale netwerk, vind ik de menselijke grondslag van deze representatie zeer interessant. Daarnaast denk ik (en met mij vele andere onderzoekers) dat het bouwen van een taak-specifieke architectuur wel degelijk voordelen heeft voor de resultaten. In dit

---

<sup>3</sup>Dit zijn zuiver muziektheoretische termen waar ik niet over wil uitweiden in deze tekst. Wie wil weten hoe deze musicologische gereedschappen zich precies verhouden tot de perceptie van intervallen, raad ik een standaard naslagwerk voor muziektheorie aan, zoals die van Theo Willemze uit 1964.



**Figuur 3:** De PHCCCF-representatie, met een dimensie voor toonhoogte (links), twee dimensies voor de chromatische cirkel (midden) en twee dimensies voor de kwintencirkel (rechts).

geval is de overeenkomst tussen representatie in het hoofd (naar Shepard) en representatie in het netwerk de voorsprong die CONCERT heeft op andere programma's.

## 2.3 CONCERT's sterke en zwakke kanten

We hebben nu genoeg middelen om te kijken naar het functioneren van CONCERT: hoe klinken de melodieën die worden geschreven? De technieken die gebruikt worden hebben hun betrouwbaarheid en bruikbaarheid grotendeels verdiend over de jaren, maar dit biedt geen garantie voor een succes in de toekomst in een nieuwe experimentele omgeving. Laten we beginnen met het maken van onderscheid tussen de doelen waarvoor CONCERT is ingezet; het programma is voor meer gebruikt dan het schrijven van melodieën in de stijl van Bach. Er werd CONCERT gevraagd noten te voorspellen na het leren van verschillende soorten invoer, zoals diatonische toonladders<sup>4</sup>, chromatische toonladders die elkaar om-en-om afwisselen (a1,b1,a2,b2,...) en toonladders die of een chromatische stap omhoog of een chromatische stap omlaag maken. Het is belangrijk om onderscheid te maken tussen deze toonladder-taken en het daadwerkelijk componeren van melodieën, omdat ze totaal verschillende doelen hebben. Je kunt niet zeggen dat een melodie geslaagd is alleen omdat hij bestaat, terwijl een juiste voorspelling wel geslaagd is omdat een specifieke noot wordt gekozen. In het tweede geval zijn er goede en foute antwoorden, in het eerste geval niet.

CONCERT heeft relatief ten opzichte van andere programma's die dezelfde taken uitvoeren, vreselijk goede statistieken (Mozer, 1994). De hierboven beschreven taken met verschillende toonladders als invoer worden haast zonder fouten geleerd door het neurale netwerk. Dit betekent in ieder geval dat CONCERT de structuur van redelijk voorspelbare reeksen noten goed kan leren en dat is hoopgevend voor het eveneens redelijk voorspelbare karakter van de muziek van Bach.

In een specifiek geval scoorde CONCERT echter minder goed. In deze taak werd het programma geacht een A-A-B-A patroon te leren (A en B staan voor respectievelijk verschillende chromatische reeksen van vijf noten; chromatisch betekent met slechts verschillen van een halve toonsafstand). Het is hier belangrijk om op te merken dat er twee informatiestromen lopen: een die slechts een noot en zijn directe voorganger beschrijft en een die beschrijft wanneer een nieuwe A of B wordt ingezet. De eerste informatiestroom verschilt in essentie niet van eerdere en was voor CONCERT geen probleem. De tweede vereist echter begrip van een meer globale logica, een die niet alleen afhangt van de directe voorganger. Hier scoorde CONCERT aanzienlijk slechter dan op alle andere.

---

<sup>4</sup>Ja, meer muziektheorie... Zie <sup>2</sup>

De melodieën in de stijl van Bach werden geschreven nadat het netwerk een aantal echte (maar gemodelleerde) werken als invoer had geanalyseerd. Ze zijn in mijn oren, en in de oren van de auteurs, niet creatief: "... *few listeners would be fooled into believing that the pieces had been composed by a human.*" (Mozer, 1994). Dit is, bovenstaande over globale structuur in gedachten, niet onverwacht: als CONCERT al moeite heeft met het leren van A-A-B-A-patronen, kan het al helemaal niet muziek schrijven in de stijl van Bach, die vol zit met globale structuur. Omdat dit knelpunt het epicentrum van mijn onderzoek is, wil ik het belang van deze globale structuur en de bijbehorende moeilijkheden voor een programma iets verder uitdiepen.

### 3 Componeren met structuur en thema's

Dit hoofdstuk is niet bedoeld als uitgebreide muziekanalyse of als duik in de wetenschap van de muziek van de grote klassieken. Echter, om duidelijk te maken wat het verband is tussen de tekortkomingen van CONCERT en door een computerprogramma gecomponeerde muziek die ik creatief kan noemen, zal ik moeten benadrukken hoe globale structuur in de praktijk nodig is om in muziek spanning en anticipatie op te bouwen. Of, misschien beter gezegd, ik zal moeten uitleggen waarom ik denk dat de muziek van CONCERT mij niet creatief in de oren klinkt en ik ontkom hierbij niet aan een kleine musicologische uitweiding.

#### 3.1 Structuur en thema's in muziek

Een goede, creatieve melodie is geen opeenvolging van mooie, losse noten. Noten klinken op zichzelf niet mooi: ze klinken mooi in hun context van andere noten (bijvoorbeeld: een C is slechts een noot, maar een C met een E en een G is een onderdeel van een harmonieus akkoord). Het zijn dan ook niet de losse noten die de spanning van een melodie dragen, maar de melodie als geheel en de manier waarop deze is opgebouwd.

Musicologen zijn het erover eens dat ritme van groot belang is voor de aantrekkingskracht van een melodie. Wanneer we een mooie en beproefde melodie nemen (zij het Jingle Bells, of een melodie uit een lied van Burt Bacharach of The Beatles) en we hem herschrijven zodat alle noten even lang duren, dan zou de spanning er vanaf zijn<sup>5</sup>. Ritme is het element dat richting geeft aan de opeenvolging van noten en het is de temporele aard van de verhouding tussen verschillende noten die ervoor zorgt dat we als luisteraars de spanning voelen en 'willen horen wat er komt'. Hiermee bedoel ik niet dat toonhoogte geen invloed heeft op eventuele spanning, maar dat het ritme ervoor zorgt dat de tonaliteit van melodie goed tot zijn recht komt. Het componeren van coherente ritmes door een computerprogramma is nog een probleem apart, en een probleem dat ik niet verder zal onderzoeken. Ik wil met het belang van ritme en ritmische coherentie slechts illustreren dat

---

<sup>5</sup>Er is me ooit op kleurrijke en memorabele wijze geïllustreerd dat melodieën waarvan de noten in een totaal ander ritme worden gespeeld onherkenbaar zijn, terwijl ze prima herkenbaar zijn wanneer noten en toonhoogte worden weggelaten en slechts het ritme wordt getikt.



muzikale spanning zich verschuilt in de verhouding tussen verschillende noten in de tijd.

Hiermee samenhangend wil ik het belang van thematische structuren in muziek benadrukken. Zowel binnen een melodie als tussen verschillende melodieën die elkaar opvolgen moeten er verbanden zijn tussen de bouwstenen (noten, groepen noten, melodieën en frases). Binnen een melodie hebben we op zijn minst een begin en een einde nodig om het natuurlijke gevoel te krijgen dat gepaard gaat met een goede melodie. Tussen melodieën die elkaar volgen kun je denken aan vraag-antwoord relaties of variaties.

Op een nog meer globaal niveau is een muzikaal thema (vaak een specifieke melodie of een deel daarvan) nodig om de aandacht van de luisteraar langdurig vast te houden. De componist van het stuk speelt met dat thema door middel van herhalingen, variaties en vele andere muzikale trucs. Zonder zulke thematische structuur klinkt een stuk als een verzameling losse frases, in plaats van als coherent geheel. De muziek van Bach en zijn tijdgenoten is zeer weloverwogen ingedeeld in delen die deze thematische structuur goed tot zijn recht laat komen en de aandacht van de luisteraar vanaf de eerste keer dat ze het stuk horen, vasthouden<sup>6</sup>.

## 3.2 Structuur en thema's in RNN's

Hoe gaan computerprogramma's om met deze belangrijke globale structuur van thema's? Programmeren zien doorgaans het belang van de muzikale inzichten en technieken wel in, maar dit betekent niet automatisch dat ze ook in staat zijn deze te integreren in de programma's die ze schrijven.

Wetenschappers die aan algoritmische compositie werken, maken, zoals gezegd, vanwege zijn voordelen veel gebruik van de connectionistische aanpak: ANN's hebben geen kennis of voorgeprogrammeerde regels nodig en zijn in staat veel statistische eigenschappen uit de invoerdata te extraheren. ANN's zijn echter niet in staat temporele informatie over globale structuren bij te houden. Een ANN kan in feite alleen noten achter elkaar plakken die gekozen worden op basis van het leerproces en de specifieke gewichtenmatrix die daaruit voortkomt. Zoals ik in hoofdstuk 2.1 heb beschreven, hebben wetenschappers om deze reden de RNN's ontwikkeld, die uitgerust zijn met

---

<sup>6</sup>Dit was ook nodig voor de muziek van Bach en zijn tijdgenoten, omdat er geen manier bestond om de muziek op te nemen. Veel stukken werden slechts een keer uitgevoerd en de componist moest dus alles uit de kast trekken om het stuk vanaf de eerste keer te laten aanspreken. Bach was hierin een meester.

een eigenschap die ze de mogelijkheid geeft temporele informatie op te slaan. Dit betekent in principe dat ze in staat zijn bij te houden waar ze zijn in het schrijven van een stuk en daarmee anticiperende elementen toe kunnen voegen aan hun composities. Er is om deze reden veel gebruikt gemaakt van RNN's in algoritmische compositie (Papadopoulos en Wiggins, 1999).

In de praktijk, echter, werkt deze manier van componeren minder goed dan je zou verwachten. Er zit namelijk een aantal conceptuele haken en ogen aan deze aanpak met RNN's. Een van de belangrijkste is dat een RNN, ondanks zijn uitrusting voor temporele informatieopslag, niet genoeg zinnigs kan zeggen over welke factoren invloed hebben op het kiezen van de juiste noten. Er is immers geen a priori manier om een RNN uit te rusten met kennis over welke informatie opgeslagen moet worden, of hoe lang informatie opgeslagen moet worden voordat het niet meer relevant is. In de woorden van Mozer: *"..., if a network is trained on sequences in which one event predicts another, the relationship is not hard to learn if the two events are separated by only a few unrelated events, but as the number of intervening events grows, a point is quickly reached where the relationship cannot be learned."* (Mozer, 1994). Dit zou erop kunnen wijzen dat de kracht van een RNN niet genoeg is voor het extraheren van structuren die zo globaal zijn als het variëren op thema's of het herhalen van eerder gebruikte melodieën.

### 3.3 Structuur en thema's in CONCERT

Mozer en Soukup hebben CONCERT ontwikkeld met de problematiek van globale structuur specifiek in gedachten. Dit blijkt onder meer uit het gebruik van een RNN. Echter, zoals gebleken is uit het voorgaande, RNN's hebben op zichzelf wellicht niet genoeg kracht om een noot-voor-noot benadering van compositie toe te passen op het componeren van gestructureerde melodieën. De melodieën van CONCERT zijn een ondersteuning van deze mogelijkheid: thema's, herhalingen en ritmische coherentie zijn volledig afwezig in de gecomponeerde melodieën.

De reden hiervoor ligt volgens mij diep in de architectuur van CONCERT. Ik denk namelijk dat de noot-voor-noot benadering van componeren het schrijven van creatieve melodieën heel moeilijk maakt. In hoofdstuk 3.1 heb ik beargumenteerd dat de aantrekkingskracht van een melodie niet ligt opgeslagen in de losse noten, maar in de manier waarop de noten zich tot elkaar verhouden. Om gebruik te maken van deze (in dit geval temporele) verhoudingen kan een componist (zij het een programma of een mens) nooit

noten een voor een achter elkaar plakken zonder te weten wat de functie van een noot is voor de noot die daarna komt (Ter vergelijking: een schrijver kan geen boek schrijven door letters een voor een achter elkaar te plakken, zonder te weten welk woord er geschreven gaat worden op het moment dat hij of zij een letter op papier zet).

De aanpassing van CONCERT die ik hierna zal beschrijven, heeft deze moeilijkheid van de noot-voor-noot benadering van componeren om thematische structuren te produceren als centraal uitgangspunt. Ik zal proberen de architectuur van CONCERT zoveel mogelijk intact laten (omdat de conceptuele benadering van de auteurs van dit programma wat mij betreft de juiste is), en alleen deze zwakte te elimineren. Omdat het in de praktijk onmogelijk zal blijken meerdere noten tegelijkertijd toe te voegen, zal ik beschrijven hoe we via andere wegen dezelfde resultaten kunnen benaderen. Nadat ik heb uitgelegd hoe mijn aanpassing in elkaar steekt, zal een evaluatie plaatsvinden.

## 4 CONCERT 2.0: mijn aanpassing

### 4.1 Complexiteit van programma's

Dat ANN's en RNN's noten een voor een toevoegen aan hun composities, is geen inherente vereiste voor hun gebruik in algoritmische compositie. De reden dat deze aanpak toch veel wordt gebruikt, is dat het een efficiënte manier van componeren is in termen van de rekenkracht die nodig is. In het algemeen is rekenkracht altijd een belangrijke factor in de computerwetenschap: grote problemen vereisen vaak een grote berekening en wanneer het vergroten van de invoer zorgt dat de berekening te groot wordt, wordt de aanpak vaak gezien als onbruikbaar. De grootte van de berekening als functie van de grootte van de invoer van een programma wordt de complexiteit van het programma genoemd. Een groter-dan-lineaire complexiteit is zeer onwenselijk.

Een andere restrictie die aan algoritmische compositie wordt opgelegd door deze complexiteitseis, is de grootte van de temporele context. Met temporele context bedoel ik het deel van de compositie dat wordt gebruikt bij de keuze voor de volgende noot. Wanneer een programmeur een RNN zo zou inrichten dat de keuze voor een noot in een compositie af zou hangen van de gehele voorafgaande compositie (i.e. alle losse noten), zou de complexiteit te groot zijn. Immers, er moet rekening gehouden worden met een steeds groter wordende context naarmate de compositie (en dus de invoer) groter wordt. Dit zou in de praktijk niet werken. Omdat deze gehele context wel belangrijk is voor het produceren van thematische structuren en ritmische coherentie, is het nodig om de context op de juiste manier in te zetten. Dit betekent dat we niet de *losse* noten als context gaan laten dienen, maar *groepen* noten.

### 4.2 Notenclusters van verschillende groottes

Door de context bij het toevoegen van een noot op te delen in relevante clusters, kan een grotere context in beschouwing genomen worden, zonder dat dit gepaard gaat met een te grote toename van de complexiteit. Ons programma kan zo de twee laatst toegevoegde noten gebruiken als context, door ze als enkel stuk invoer te gebruiken. Of de laatste vier noten, of de laatste tien. Het verschil met de oorspronkelijke versie van CONCERT is dat het netwerk door het ter gebruik stellen van al die clusters nu beschikt over

een veel efficiëntere manier om zijn informatiebron te gebruiken. De RNN van CONCERT was hiervoor niet krachtig genoeg.

Wat we idealiter ook zouden willen, is de mogelijkheid is om meerdere noten tegelijk toe te voegen. Als we op basis van een context van (bijvoorbeeld) vijf noten ook tegelijk vijf noten zouden kunnen toevoegen, zou het netwerk veel potentie hebben om interessante patronen van verschillende groottes te leren. Er schuilt hier echter een gigantisch computationeel addertje onder het gras. Bij het toevoegen van groepen noten lopen we kans om uit de bocht te vliegen wat betreft de grootte van de benodigde berekening. Wanneer we namelijk voor elke combinatie van noten die we mogelijk gaan toevoegen een aparte uitvoerknoop moeten maken die de combinatie representeert, hebben we astronomisch veel uitvoerknopen nodig. Stel, we willen in een keer vijf noten toevoegen aan onze compositie. Met de 49 verschillende toonhoogtes die volgens Mozer nodig zijn voor het componeren van genoeg variërende melodieën (Mozer, 1994) zijn er ( $49^5 =$ ) 282.475.249 unieke combinaties van vijf noten (bijvoorbeeld G-A-Bes-F-A, B,B,B,D-Ges etc.) en daarmee zijn er precies zoveel uitvoerknopen nodig. Naast het feit dat een computerprogramma zoveel mogelijkheden niet snel genoeg kan overwegen, leidt zo'n groot aantal uitvoerknopen er ook toe dat er geen zinnig onderscheid gemaakt kan worden tussen twee verschillende opties. Wanneer er zo'n groot aantal uitvoerknopen tot de keuzemogelijkheden behoort, zal geen enkele knoop significant geschikter zijn voor een specifieke situatie dan een andere. We zitten daarmee vast aan de representatie van de uitvoerknopen als noot, in plaats van de representatie als notencluster zoals we die zo graag zouden willen.

Om het effect van het tegelijkertijd toevoegen van meerdere noten alsnog te benaderen, zullen we specifiek moeten zijn in wat onze context tijdens elke tijdstap is en wanneer we deze context laten veranderen. De context bepaalt namelijk, samen met de gewichtenmatrix, de volgende noot. Ik leg dit uit aan de hand van een voorbeeld. Wanneer we een noot hebben toegevoegd aan de hand van een context van vijf noten (een cluster van de vijf vorige noten die zijn gecomponeerd), zullen we de gewichtenmatrix en de context *moeten* updaten. Wanneer we dit niet zouden doen zou dezelfde noot op basis van dezelfde kansverdeling vijf keer opnieuw gekozen worden. Daarom zal de net gekozen noot voor de volgende stap aan het einde van de context geplakt worden en zal de gewichtenmatrix worden geüdate. De context bestaat dan bij het toevoegen van de volgende noot uit zes noten. Omdat we het in een keer toevoegen van meerdere noten wilden benaderen, laten we de context af en toe drastischer veranderen. Bij het toevoegen van vijf noten zullen we

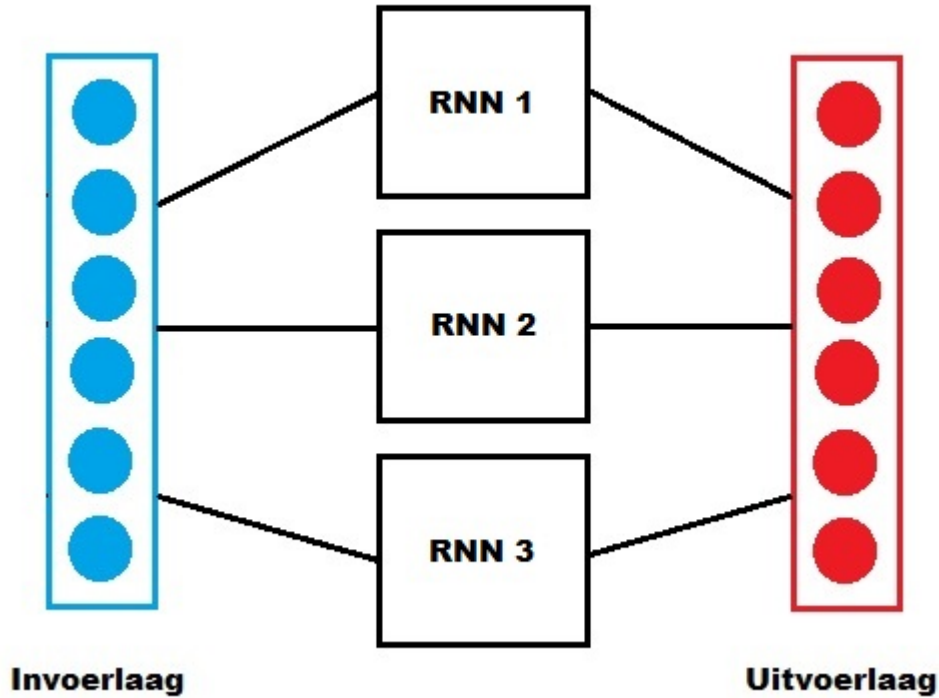
deze stap herhalen tot er vijf noten zijn toegevoegd (en de context uit tien noten bestaat). Dan zal de oorspronkelijke context van vijf noten worden gewist en zullen de volgende vijf noten worden gekozen op basis van een verse context, namelijk de vijf noten die we net hebben toegevoegd.

Dat we gebruik maken van een specifieke contextgrootte betekent echter nog niet dat we in staat zijn meer informatie te gebruiken dan een andere RNN met een andere context. Elke noot wordt immers in beide gevallen gekozen met een van tevoren vastgestelde context. We willen bij het toevoegen van een noot eigenlijk op hetzelfde moment toegang hebben tot meerdere contextgroottes zodat er meer temporele relaties tussen de noten beschikbaar zijn voor het programma.

### 4.3 Parallele RNN's

Om gebruik te kunnen maken van meer clusters noten dan alleen van die van vijf, wil ik meerdere RNN's (met de PHCCCF-representatie en andere architectonische eigenschappen van de oude CONCERT intact) parallel laten rekenen, waarbij elke RNN verantwoordelijk is voor een unieke contextgrootte. Zo is er een RNN die slechts een noot als context meeneemt (en dus na elke stap met een verse context begint), een RNN die twee noten meeneemt, een RNN die drie noten meeneemt, enzovoorts. Omdat het aantal clusters dat we kunnen gebruiken lineair toeneemt met de groei van de invoer, kunnen we een groot aantal RNN's parallel laten rekenen. We kunnen in de praktijk testen wat het meest gunstige aantal is. Zie figuur 4 voor een voorstelling van een netwerk van parallele RNN's.

Het is belangrijk om te begrijpen dat de uiteindelijke keuze van welke noot aan de compositie wordt toegevoegd pas plaatsvindt nadat elke RNN zijn eigen selectie heeft gemaakt in zijn uitvoerlaag. Elke RNN produceert een noot op basis van zijn eigen gewichtenmatrix en kansverdeling en daarna draagt elke RNN zijn noot voor bij de eindverkiezing die plaatsvindt in de algemene uitvoerlaag van het programma. In de uitvoerlaag representeert elke knoop de noot van een specifieke RNN. Er is dus een aantal uitvoerknoopen dat gelijk is aan het aantal noten dat tot nu toe gecomponeerd is (met als plafond het aantal RNN's dat we willen gebruiken). Er wordt uiteindelijk gekozen voor de noot die van alle noten de beste waarde had (binnen zijn eigen kansverdeling van zijn eigen RNN). Bijvoorbeeld, als we drie RNN's gebruiken (met respectievelijk een context van een, twee en drie noten), draagt elke RNN zijn eigen noot voor aan de eindverkiezing (bijvoorbeeld repsectie-



**Figuur 4:** Een netwerk van drie parallelle RNN's. RNN1 gebruikt een context van een noot, RNN2 een context van twee noten en RNN3 een context van drie noten. De invoerlaag dient als algemene representatie van de invoerdata, zonder specifieke context. De uitvoerlaag is waar de uiteindelijke selectie plaatsvindt, en bevat de beste kandidaat van elke losse RNN. Elke RNN heeft ook zijn eigen, interne in- en uitvoerlaag

velijk C, Bes en Es). Van deze drie noten wordt in de algemene uitvoerlaag van het programma de noot gekozen die de hoogste waarde had (De C in de kansverdeling van RNN1, de Bes in de kansverdeling van RNN2 of de Es in de kansverdeling van RNN3). Op deze manier wordt de kandidaat die het meest geschikt is voor de specifieke situatie waar de compositie zich op dat moment in bevindt op basis van alle contextgroottes die we in beschouwing willen nemen, gekozen als nieuwe noot.

## 5 Potentie en Discussie

### 5.1 CONCERT 2.0 in de praktijk

Wat ik wil bereiken met CONCERT 2.0 is dat het bij het componeren van zijn melodiën de touwtjes zoveel mogelijk zelf in handen heeft. Hiermee bedoel ik dat we als architect het programma zo min mogelijk probleemspecifieke informatie meegeven. Hoe meer we namelijk zelf (als mensen) bepalen voor de manier waarop CONCERT zijn composities maakt, hoe minder we van het programma kunnen zeggen dat het erin geslaagd is in het schrijven van (al dan niet) creatieve melodieën. Van een programma dat moet omgaan met een strenge (menselijke en probleemspecifieke) afbakening van zijn mogelijkheden kan immers moeilijk gezegd worden dat de composities door hemzelf zijn geschreven.

Ik denk dat CONCERT 2.0 in dit opzicht potentie heeft: als architect van een programma hoeven we niet te weten waar thema's zich in de (invoer)muziek bevinden, hoe groot ze zijn of wat voor functie ze dragen voor de muziek, zolang we ons programma uitrusten met een leerprocedure die deze informatie zelf kan infereren. Omdat CONCERT 2.0 beschikt over een krachtige compositieprocedure die gebruik kan maken van een willekeurig aantal notenclusters om thematische structuren te infereren verwacht ik dat een goed leertraject kan leiden tot goede resultaten. De RNN's die parallel aan het rekenen zijn zorgen voor een (deel)resultaat dat optimaal is voor elke gegeven context en van al deze resultaten wordt de meest geschikte toegevoegd aan de compositie.

Voordat ik hierover al te sterke voorspellingen maak, moet ik het gevaar benoemen van de mogelijkheid dat het Backpropagation-algoritme dat gebruikt wordt om de gewichtenmatrix te updaten niet sterk genoeg is om deze berg informatie zelfstandig naar iets bruikbaar te kunnen kneden. Het zou kunnen dat we het programma een handje moeten helpen door de grootte van de invloed van alle verschillende notenclusters handmatig bij te stellen. Het is immers onwaarschijnlijk dat Bach zijn muziek schreef met thema's bestaande uit 25 noten, en waarschijnlijker dat veel van deze thema's bestaan uit drie of vier noten. Als de thema's in de invoermuziek inderdaad bestaan uit (voornamelijk) drie noten, moet de RNN die verantwoordelijk is voor de context van drie noten meer inbreng krijgen in de eindverkiezing dan andere RNN's (een verkiezingsbias). Een test in de praktijk zal moeten uitwijzen wat hiervoor de beste strategie is: misschien is een uniforme



invloedsverdeling de beste, misschien hebben we een grote bias nodig voor goede resultaten. Ik hoop dat het leeralgoritme krachtig genoeg is om een menselijk ingrijpen overbodig te maken. Wat helemaal fantastisch zou zijn, is als er een manier gevonden wordt om het programma zelf een optimale invloedsverdeling te laten leren. Op deze manier wordt de uniformiteit van de invloed van de parallelle berekeningen doorbroken zonder dat we handmatig moeten ingrijpen.

Een ander probleem, waarvan de significantie eveneens pas bevestigd danwel ontkracht kan worden na het testen van CONCERT 2.0 in de praktijk, is dat het paradoxaal is om een coherente compositie met globale structuur te verwachten van een programma dat op zoveel (semi-)onafhankelijke niveaus tegelijk aan het werk is. De verschillende RNN's die thematische structuren uit de notenclusters van verschillende groottes moeten infereren, hebben allemaal een kans om gekozen te worden voor de eervolle taak van het toevoegen van de volgende noot. Welke noot er gekozen gaat worden hangt van veel factoren af, waaronder de noten die tot nu toe gecomponeerd zijn en de invoerdata waarop het netwerk is getraind. Het zou dus kunnen dat er drie opeenvolgende noten worden gekozen die geen verband met elkaar houden in termen van globale structuren. Ik verwacht, vanwege de invloedsverdeling van de verschillende RNN's, dat dit niet zal gebeuren: een 'belangrijkere' RNN heeft meer invloed op de compositie dan een minder belangrijke. De praktijk zal hierover uitsluitsel moeten geven.

## 5.2 Filosofische overwegingen voor vervolgonderzoek

Tot slot wil ik kort een aantal overwegingen bespreken die het onderzoeken van computationele creativiteit zullen stimuleren. Als eerste en belangrijkste wil ik bekijken wat het betekent om bezig te zijn met het nabootsen van creativiteit met programma's. Gedurende de laatste paar weken heb ik namelijk meerdere malen en van verschillende mensen iets van de volgende vorm gehoord: "Hoe zou je creativiteit kunnen programmeren? Als de uitvoer van een (deterministisch) computerprogramma komt, is er toch per definitie niets creatiefs aan?" Dit zijn moeilijke vragen waarop ik geen eenduidig antwoord kan geven. Mocht er zo'n antwoord zijn, dan denk ik dat die gevonden kan worden door dieper in te gaan op wat de processen in het menselijk cognitieve systeem zijn die creatieve uitspattingen veroorzaken of vergezellen. Ik zie op papier geen reden om aan te nemen dat computationele creativiteit niet mogelijk is.

Papadopoulos en Wiggins (1999) stellen een soortgelijke vraag: "Do we want to simulate human creativity or the results of it?": zijn we tevreden met een deterministisch geproduceerde uitvoer die sterke gelijkenissen vertoont met creatieve uitingen die door mensen zijn gemaakt, of zit het creatieve aspect in de processen die aan de uitvoer vooraf gaan? In mijn onderzoek onderschrijf ik optie 1, omdat in mijn ogen menselijke creativiteit nog lang niet benaderd is door computerprogramma's. We zijn momenteel zo ver verwijderd van creatieve programma's dat het in mijn ogen te vroeg is de onderliggende mechanismes die ze gebruiken (inclusief eventuele overtuigingen, intenties en bedoelingen van het programma) het bepalende criterium te maken. Dit standpunt geeft mij de mogelijkheid een vrije interpretatie van creativiteit te gebruiken. Onderzoekers zijn het er in de praktijk echter niet over eens of een objectieve maatstaf voor potentieel creatieve uitvoer zou moeten bestaan en wat die maatstaf dan zou moeten zijn (Papadopoulos en Wiggins, 1999).

Een overweging die meegenomen moet worden door onderzoekers in het veld van algoritmische compositie, is het verschil tussen componeren en improviseren. Wat me duidelijk is geworden tijdens het onderzoek, is dat wat CONCERT, CONCERT 2.0 en eigenlijk alle programma's die de connectionistische aanpak gebruiken doen, niet veel lijkt op componeren zoals dit wordt gedaan door mensen. Een componist schrijft een stuk muziek met het overzicht van een 3<sup>e</sup> persoon, een 'bird's-eye view'. Hij heeft de mogelijkheid noten, thema's of zelfs hele delen van de compositie achteraf naar gelang aan te passen. Computerprogramma's hebben deze mogelijkheid niet: als er een stuk wordt geschreven, kan het niet meer worden veranderd. Wat deze programma's doen, lijkt in feite meer op *improviseren*. Een muzikant die improviseert, heeft ook maar een kans om de juiste noten te spelen en wat er gespeeld is, kan niet terug genomen worden. Het is het verschil tussen componeren *van binnenuit* (live, real-time) en componeren *van buitenaf*.

Al met al lijkt het duidelijk dat er nog een hoop te doen is in computationele creativiteit en algoritmische compositie. Zelfs met de potentie die CONCERT 2.0 heeft verwacht ik niet dat we zijn composities zullen verwarren met composities die door mensen zijn geschreven; zo ver zijn we nog niet. De connectionistische aanpak die ik heb gebruikt in mijn onderzoek is interessant door de modellering van het menselijke brein die aan de methode ten grondslag ligt. Ik denk echter dat hybride systemen, die meerdere methoden combineren, nog meer potentie hebben om menselijke creativiteit te benaderen. Immers, al deze systemen ontleen hun bestaansrecht aan een

aspect van menselijke cognitie en het is juist de combinatie van al deze aspecten die de menselijke cognitie zijn creatieve potentie geeft. Wanneer we dus efficiënte methoden vinden om meerdere methoden te combineren zullen de mogelijkheden zich snel verspreiden. In de tussentijd zal ik nog even naar Bach blijven luisteren.

## 6 Bibliografie

Boden, M., Dimensions of creativity, 1996, MIT press, Cambridge, USA

Churchland, P, Sejnowski, Blending Computational and Experimental Neuroscience, 2016, *Nature Reviews Neuroscience*, [www.nature.com](http://www.nature.com)

Cope, D., Experiments in Musical Intelligence, 1987, *Proceedings of the International Computer Music Conference*, San Fransisco, USA

Denker, J., Schwartz, D., Wittner, B., Solla, S., Howard, R., Jackel, L., Hopfield, J., Large automatic learning, Rule extraction, and Generalization, 1987, *Complex Systems 1*, p877-922, Champaign, USA

Ebcioğlu, K., An expert system for harmonizing chorales in the style of J.S. Bach, 1990, *Computer Music Journal*, Cambridge, USA

Moore, G.M., Lithography and the Future of Moore's Law, 1995, *Proceedings of SPIE*, Santa Clara, USA

Mozer, M.C., Induction of multiscale temporal structure, 1992, *Advances in Neural Information Processing Systems (NIPS)*, p275-282, San Francisco USA

Mozer, M.C., Neural Network Music Composition by Prediction: Exploring the Benefits of Psychoacoustic Constraints and Multi-scale Processing, 1994, *Connection Science, Vol 6*, p227-260, Cambridge USA

Mozer, M.C., Soukup, T., Connectionist Music Composition Based on Melodic and Stylistic Constraints, 1991, *Advances in Neural Information*

*Processing Systems (NIPS), p789-796, San Fransisco USA*

Papadopoulos, G., Wiggins, G, AI Methodes for Algorithmic Composition: A Survey, a Critical View and Future Prospects, 1999, *AISB Symposium on Musical Creativity*

Shepard, R.N., Geometrical approximations to the structure of musical pitch, 1982, *Psychological Review*, 89, p305-333