

UNIVERSIDADE ESTADUAL PAULISTA

“Julio de Mesquita Filho”

Instituto de Artes - Campus São Paulo

Henri Augusto Bisognini

Composição Musical e Inteligência Artificial: um jogo-pesquisa

São Paulo

2022

Henri Augusto Bisognini

Composição Musical e Inteligência Artificial: um jogo-pesquisa

Trabalho de Conclusão de Curso apresentado
por Henri Augusto Bisognini, graduando em
Bacharelado em Composição Musical na Uni-
versidade Estadual Paulista “Júlio de Mes-
quita Filho” - UNESP.

Orientador: Alexandre Roberto Lunsqui

São Paulo
2022

Ficha catalográfica desenvolvida pelo Serviço de Biblioteca e Documentação do Instituto de Artes da Unesp. Dados fornecidos pelo autor.

B622c	Bisognini, Henri Augusto, 1993- Composição musical e inteligência artificial : um jogo-pesquisa / Henri Augusto Bisognini. - São Paulo, 2022. 70 f. : il. color. + anexos Orientador: Prof. Dr. Alexandre Roberto Lunsqui Trabalho de Conclusão de Curso (Bacharelado em Música) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Instituto de Artes 1. Composição musical por computador. 2. Inteligência artificial - Aplicações musicais. 3. Jogos - Processamento de dados. 4. Música e Tecnologia. I. Lunsqui, Alexandre Roberto. II. Universidade Estadual Paulista, Instituto de Artes. III. Título. CDD 781.34
-------	--

Bibliotecária responsável: Laura M. de Andrade - CRB/8 8666

Henri Augusto Bisognini

Composição Musical e Inteligência Artificial: um jogo-pesquisa

Trabalho de Conclusão de Curso apresentado
por Henri Augusto Bisognini, graduando em
Bacharelado em Composição Musical na Uni-
versidade Estadual Paulista “Júlio de Mes-
quita Filho” - UNESP.

Data da Defesa: 13 / 01 / 2022 Nota: _____

Banca examinadora:

Alexandre Roberto Lunsqui
Orientador

Maurício De Bonis
Convidado

São Paulo
2022

Agradecimentos

Agradeço à minha mãe, Maria da Claridade Janicas Bisognini, e ao meu pai, Evaristo Bisognini Junior, por todo o amor e apoio que me deram durante toda a minha vida. Sobre seus ombros de gigantes pude enxergar muito mais longe do que se estivesse com os pés no chão.

Agradeço à Karoline Cassiano pela ajuda durante a elaboração deste trabalho, toda a minha graduação e todos os meus projetos mirabolantes de arte.

Agradeço ao meu orientador, Alexandre Lunsqui, pelo apoio, ajuda, compreensão e paciência durante a realização deste trabalho e ao membro convidado da banca de defesa, Maurício De Bonis, por enriquecer grandemente nosso aprendizado com suas observações.

Agradeço a todos os meus professores por terem me inspirado a cada passo durante minha infinita jornada de aprendizado.

Agradeço a todos os meus amigos e familiares por compartilhar as dores, risos, inquietações e alegrias da vida.

Resumo

Neste trabalho investigaremos os usos criativos de Inteligência Artificial aplicados à música, em especial à composição musical. Tais sistemas artificiais são capazes de tomar decisões musicais por nós. Realizaremos uma revisão bibliográfica sobre as técnicas de *IAs* destinadas à música e seus usos por artistas a fim de compreender melhor a atratividade destes sistemas, que são popularmente conhecidos por substituir seres humanos nas mais diversas atividades, numa tentativa de responder às diversas questões, como, por exemplo: “desejamos que estes sistemas tomem decisões musicais por nós”? Também desenharemos e implementaremos um jogo de composição musical feito com Inteligência Artificial a fim de explorar as habilidades técnico-musicais da *Magenta*, *IA* criada pelo *Google Brain Team*. Por meio de um relato de experiência, pretendemos entender o quanto estas ferramentas já estão (ou não) adequadas a facilitar, ampliar e potencializar o trabalho composicional. Adicionalmente, buscaremos elaborar o texto como uma introdução aos conceitos de *IA* para músicos que não têm um *background* em ciências de dados e técnicas de aprendizado de máquina.

Palavras-chaves: música, composição musical, arte, criação, inteligência artificial, aprendizado de máquina, jogos, jogos de cartas, Google Magenta

Abstract

In this work we will investigate creative uses of Artificial Intelligence in music, with emphasis on composition. Such systems are able to make musical decisions for us. We'll realize a literature review about *AI* techniques applied to music and it's uses by artists in order to seek a better comprehension about the attractiveness of such systems, which are popularly known to replace humans in a wide range of activities, in an effort to answer diverse questions such as: "do we desire systems that make musical decisions for us?". We'll also design and implement a musical composition card game made with *AI* with the goal of exploring the technical musical abilities of *Magenta*, an *AI* created by the *Google Brain Team*. Finally, by reporting our experience, we aim to understand how those tools are adequate (or not) to facilitate and amplify the musical compositional work. Additionally, we'll attempt to write this text as an introduction to the concepts of *AI*, intended for musicians which do not have a background in data science and machine learning techniques.

Keywords: music, musical composition, art, creation, artificial intelligence, machine learning, games, card games, Google Magenta

Sumário

	Sumário	7
1	Introdução e justificativa: Inteligência Artificial e tomada de decisões	9
2	Metodologia	11
3	algoritmos e Inteligências Artificiais	12
3.1	Breve introdução às redes neurais	14
3.2	Treinando uma rede neural	15
4	“Habilidades” musicais realizadas com Inteligência Artificial	19
4.1	Criação direta	19
4.2	Derivação	21
4.3	Acompanhamento <i>off-line</i>	21
4.4	Improvisação	21
4.5	Mesclagem	22
4.6	Análise	22
4.7	Harmonização	23
4.8	Orquestração	23
4.9	Expressividade humana	24
4.10	Reconhecimento ótico de partituras	24
4.11	Acompanhamento de partitura (<i>Score Following</i>)	25
4.12	Hiper-instrumentos	25
5	Jogo: <i>Carmim</i>	26
5.1	Visão geral da interface	27
5.2	As cartas	28
5.2.1	Cartas de sequência: melodia, bateria, geradoras e não-geradoras	28
5.2.2	Carta de variação	29
5.2.3	Carta de continuação	30
5.2.4	Carta de interpolação	31
5.2.5	Carta de edição	32
5.3	Playtests	34
5.3.1	Playtest 1	34
5.3.2	Playtest 2	34
5.3.3	Playtest 3	35
5.3.4	Playtest 4	35
5.3.5	Playtest 5	36
5.3.6	Playtest 6	36

5.3.7	Playtest 7	37
5.3.8	Playtest 8	37
5.3.9	Playtest 9	38
5.4	Investigação sobre IAs e composição através do Jogo <i>Carmim</i>	38
5.4.1	As funcionalidades de IA implementadas no jogo	38
5.4.2	Funcionalidades não fornecidas pela biblioteca	39
6	Magenta.js	41
6.0.1	Breve introdução matemática à Interpolação Linear	41
6.0.2	<i>Variational Auto Encoders</i>	44
6.1	Amostragem	47
6.2	Interpolação	48
6.3	Variação	49
6.4	Redes recursivas e a habilidade de continuação	49
7	Uso das ferramentas de IA fora da academia	51
7.1	YATCH	51
7.2	Shadow Planet	53
7.3	Hello World e FlowMachines	54
7.4	O problema do licenciamento	55
7.5	Considerações sobre a adoção e impacto da IAs por artistas fora do circuito acadêmico	56
7.5.1	O potencial de apropriação de todo software	58
8	Conclusão e direcionamentos para o futuro	59
	Referências	60
9	Apêndice	64
9.1	Excerto criado com o jogo Carmim	64

1 Introdução e justificativa: Inteligência Artificial e tomada de decisões

Em 11 de novembro de 2019 assisti a uma entrevista com Yuval Noah Harari no programa Roda Viva da TV Cultura (YUVAL NOAH HARARI, 2019). Uma das perguntas ao entrevistado foi sobre qual seria o sentido da vida em um mundo com Inteligência Artificial. Sua resposta foi inquietante: disse que toda a nossa concepção de vida é baseada em um drama de tomada de decisões.

Vejo que esta resposta abrange tudo que diz respeito à nossa individualidade. A história de vida que contamos é a história de nossas decisões. Ela é escrita pela interação entre as variáveis que estão fora de nosso controle com as que controlamos, ou seja, entre os percalços da vida, nosso intelecto e nossa personalidade. Propriedades cognitivas estas que são como o nosso “motor de tomada de decisões”. Como será nossa vida quando sistemas artificiais forem mais capazes do que nós mesmos para tomar decisões sobre nossa vida? Ele disse ainda que as respostas cabem “ao indivíduo, ao filósofo, ao poeta, ao artista”.

Neste texto extrapolamos a questão: como isto influencia nossa visão sobre arte e Inteligência Artificial? Especificamente, quais implicações isto tem para a área de composição? Compor é, de certo modo, um processo de tomada de decisões. Afinal, o que é compor senão tomar uma série de decisões sobre altura, ritmo, intensidade e timbre? Claro, estes parâmetros se desdobram em níveis mais complexos como motivo, melodia, frase, forma, textura, articulação, etc. Decidir como usar, não usar, combinar e estruturar estes fatores é a quintessência do trabalho composicional.

Então qual é a nossa visão sobre criar uma Inteligência Artificial que toma **decisões composicionais** pelo compositor? Mesmo utilizando um processo aleatório ou fora de seu controle para criar, isto é uma escolha: a escolha do que controlar e o que não controlar.

Pode-se cair na seguinte questão: se um compositor cria/treina uma *IA* para tomar decisões composicionais por ele, a quem pertence a autoria da obra? Isto é compor ou é alguma outra coisa nova? Independentemente da resposta, ainda resta a dúvida: ele de fato **quer** que uma Inteligência Artificial tome algumas decisões por ele?

Vem à mente o “Musikalisches Würfelspiel” (“*Jogo de Dados Musical*”) de Mozart. Esta, como tantas outras peças, é elaborada pelo compositor como um **sistema** para a criação de diferentes instâncias da obra. O mesmo poderia ser dito sobre outras composições feitas com outros sistemas (sejam eles feitos para uma única obra ou um sistema compartilhado) e até mesmo para sistemas que envolvem o acaso, como em obras de John Cage. A composição está na criação do sistema (até no caso mais extremo do sistema do não-controle, onde TUDO na peça é aleatório).

Tais sistemas apesar de mais simples não deixam de ser sistemas/algoritmos criados como ferramentas composicionais. Neste aspecto são semelhantes. Entretanto tais siste-

mas jamais oferecerão as possibilidades que as *IAs* nos trazem. Uma *IA* pode analisar a obra de Bach e aprender a harmonizar como Bach. De fato este foi um *Doodle do Google* onde o usuário escrevia uma breve melodia de 2 compassos e a *IA* a harmonizava.

As possibilidades se tornam mais interessantes ao se imaginar uma Inteligência Artificial integrada em um software de composição. Seria algo como um assistente de composição artificial: pode-se treinar uma *IA* para, dada uma melodia, o software criar uma variação sobre ela. Para tal bastaria selecionar a melodia e dar 1 ou 2 cliques. Poderia, com igual facilidade, solicitar 10 melodias novas que, de alguma maneira, têm elementos da música na qual se está trabalhando de modo que estas melodias façam sentido na obra. Poderia ainda dar sugestões como “Para que a harmonia fique interessante experimente introduzir novas classes de alturas nesta seção da peça”. Já existem *IAs* que conseguem *criar composições inteiras com um clique*. Uma *IA* poderia até dizer se uma música (ou excerto) será ou não interessante para determinado público-alvo.

O que observamos então é que a importância da tomada de decisões pelo compositor pode parecer cada vez mais ofuscada, mesmo que ele seja usuário de uma *IA* e assuma um papel coordenador da criação ao usar a ferramenta, ao dar os comandos de maneira arbitrária. Ou seja, por mais que esteja decidindo de maneira *macro* como usar a *IA* ou que esteja usando sua percepção para julgar qual das 100 composições que requisitou ao computador soa melhor, pode-se arguir que seu papel de tomar decisões musicais se tornou menos importante. Isto parece tirar do compositor uma espécie de responsabilidade de artesão que, utilizando uma técnica aprimorada, idealiza e molda do começo ao fim sua obra, controlando cada mínimo detalhe.

Lembro que os criadores da *AIVA*, plataforma que consegue criar composições inteiras com literalmente 1 clique, divulgaram seu serviço digital em 2 grupos de rede social dos quais participo: um de compositores e outro de desenvolvedores de jogos. No grupo de desenvolvedores de jogos as respostas foram mistas porém geralmente positivas: pode-se com muita facilidade obter trilhas sonoras baratas e com poucos cliques. Por outro lado, a reação foi bastante negativa no grupo de compositores. Alguns sentiam que estavam sendo literalmente substituídos (e sentiram-se ofendidos pela oferta do serviço justamente num grupo de compositores) enquanto outros entendem que o serviço (pelo menos no estado atual da ferramenta) não conseguiria substituir um compositor e/ou até que tal sistema poderia auxiliar um compositor ao invés de substituí-lo, proporcionando ideias com as quais trabalhar, novas ferramentas, etc. Porém, ainda assim diversos membros do grupo não viram muita utilidade em uma ferramenta que tira do compositor justamente a sua principal função: a tomada de decisões.

Fica então o questionamento: que faremos com esta tecnologia¹? O quão a desejamos? Quem a deseja e para que fim? Por exemplo: como compositor, desejaria ter como ferra-

¹ Harari deixa no ar a ideia de que os especialistas criam a tecnologia, mas o que fazer com ela é uma decisão política e/ou individual, subjetiva (dependendo de como esta tecnologia pode afetar a vida das pessoas).

menta um assistente de Inteligência Artificial integrado no meu software de composição preferido, que poderia usar ou não para me auxiliar conforme fosse conveniente. Outros criadores que utilizam música em suas mídias poderiam se beneficiar de *IAs* que criem música de maneira custo-eficiente? Que pessoas têm interesse em ouvir músicas criadas por *IAs*? (e quais *IAs*?)

O desenvolvedor independente de jogos provavelmente verá utilidade em ter uma trilha sonora completa por um preço barato e com agilidade. Talvez esta ferramenta em particular não seja tão interessante para alguns compositores mas muito atrativa para outros. Para os músicos, em especial os compositores, o que existe (ou poderia existir) em termos de Inteligência Artificial que poderia os auxiliar em tomar decisões criativas ao invés de tomá-las em seu lugar? Que ferramentas teriam o poder de ampliar seus horizontes criativos?

Provavelmente estas perguntas continuarão sem respostas (ou respostas satisfatórias) por muitos anos e, certamente, não será este trabalho que trará qualquer resposta definitiva, se é que tal consenso um dia existirá.

Entretanto, buscaremos dar um passo em direção a uma maior compreensão da relação entre tais ferramentas e os artistas, além de investigar as habilidades musicais de diferentes sistemas artificiais a fim de entender como eles afetam e/ou poderão afetar o trabalho composicional musical. Os resultados desta investigação poderão guiar futuras pesquisas e criação de novas ferramentas para preencher eventuais lacunas no conjunto de habilidades das *IAs* voltadas para música.

2 Metodologia

Naturalmente, ao investigar um tema e questões como estas, confronta-se com um universo de tamanho imensurável de possibilidades, questionamentos, realidades, conhecimento e experiências. De modo a estruturar e guiar a pesquisa com clareza e, ao mesmo tempo, mantê-la aberta às variabilidades, subjetividades e complexidades das inquirições artísticas, nos apoiamos nas ideias de [Fortin e Gosselin \(2014\)](#). Optamos por uma pesquisa exploratória em vez de usar um modelo tradicional baseado em hipótese, pois julgamos ser uma melhor opção para nos guiar em direção às elucidações sobre as questões de natureza social e artística apresentadas na introdução.

Como metodologia combinaremos uma revisão bibliográfica com um relato de experiência de criação de um jogo de composição musical feito com Inteligência Artificial. A revisão bibliográfica nos fornecerá uma visão objetiva das ferramentas de *IA* existentes na atualidade e das discussões em torno delas. O relato sobre o jogo, por outro lado, talvez nos permitirá perceber o que não está presente por escrito na bibliografia, ou seja, com ele esperamos nos aproximar das questões mais subjetivas sobre as relações entre artistas e estas tecnologias (subjetividade esta da qual é impossível fugir dada a natureza social e

artística dos questionamentos).

Ainda a fim de tentar clarear as questões sobre as relações de tais ferramentas com o meio artístico, paralelamente à revisão bibliográfica, pesquisaremos os usos criativos das ferramentas de *IA* por artistas fora do circuito das instituições de pesquisa e nos debruçaremos sobre o que eles têm a dizer sobre suas experiências.

Finalmente, como um objetivo complementar, buscaremos que o texto funcione como uma espécie de *introdução simplificada* aos conceitos de Inteligência Artificial destinada à músicos sem conhecimento técnico sobre *ciência de dados* e *aprendizado de máquina*. Assim o faremos devido à escassez de trabalhos sobre o tema no repositório do Instituto de Artes da UNESP e também como medida para uma possível capacitação de futuros pesquisadores cuja vontade é continuar a exploração deste tema.

3 algoritmos e Inteligências Artificiais

É importante fazer uma distinção entre algoritmos tradicionais e algoritmos de Inteligência Artificial. Ao escrever um algoritmo tradicional, o programador dá ao computador instruções exatas e domínio-específicas do que deve ser feito. Por exemplo: ao fazer um algoritmo que gera melodias, o programador cria no código todas as regras musicais que governam o gerador de melodias.

Um algoritmo de Inteligência Artificial é muito mais genérico. Veja a definição abaixo:

algoritmos de aprendizado de máquina constroem um modelo baseado em dados de amostra, conhecidos como dados de treinamento, a fim de fazer previsões ou decisões sem ser explicitamente programados para tal.² (MACHINE..., 2021)

Os algoritmos de Inteligência Artificial são por definição genéricos o suficiente para que se possa utilizá-los para diversos fins sem a necessidade de dar todas as instruções específicas do domínio na qual estão sendo aplicados³. Isto não significa que não se pode usar nenhum conhecimento de outras áreas ao criar uma *IA*, mas significa que o algoritmo encontrará soluções e caminhos para os desafios em questão de maneira autônoma extrapolando o conhecimento domínio-específico que foi utilizado em sua criação.

Neste sentido, criar um algoritmo “tradicional” para fazer música pode ser visto como algo mais próximo de uma composição tradicional (comparado a um algoritmo de aprendizado de máquina) uma vez que seria uma sistematização de regras de domínio teórico-musical que o compositor explicitamente escolhe.

² Original, em inglês: “Machine learning algorithms build a model based on sample data, known as training data, in order to make predictions or decisions without being explicitly programmed to do so.”

³ Naturalmente domínios diferentes são abordados de maneiras mais eficientes com tipos diferentes de inteligências artificiais. Por exemplo: as redes neurais recursivas são melhores candidatas para lidar com dados sequenciais como a música.

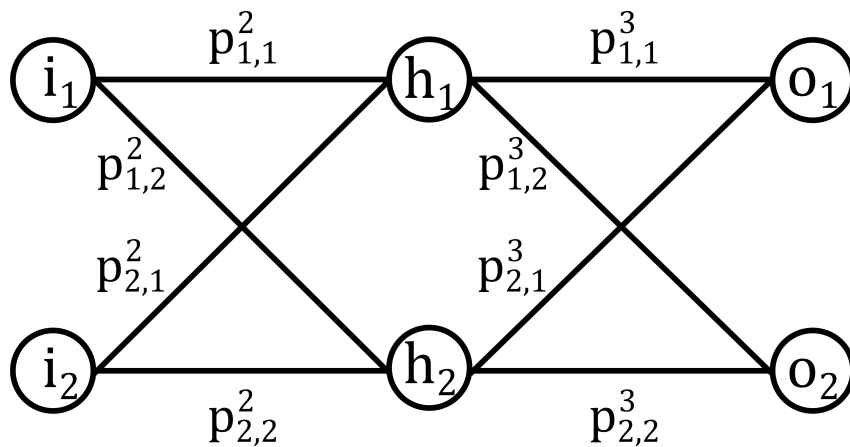
Ao treinar uma rede neural utiliza-se um algoritmo mais genérico que é capaz de inferir suas próprias regras recorrendo à análise de uma grande quantidade de dados. Inclusive estas regras geralmente tomam forma de funções matemáticas extremamente complexas que um ser humano não conseguiria entender. Utilizar uma Inteligência Artificial, portanto, seria menos um processo de criação de regras musicais e mais um processo de criação e uso de técnicas de aprendizado de máquina, por mais que algum conhecimento de domínio musical ainda seja requerido pela parte responsável pelo treinamento da *IA*.

3.1 Breve introdução às redes neurais

A fim de proporcionar uma intuição básica de como redes neurais funcionam, faremos uma introdução bastante simplificada dos seus conceitos básicos. Aqui nos atentaremos a uma visão *macro*, com enfoque em questões conceituais fundamentais, para perseguir o objetivo de permitir ao leitor, através desta intuição, entender de maneira geral como algumas operações são feitas via Inteligência Artificial. Simplificaremos e omitiremos muitos detalhes técnicos importantes da matemática que seria necessária para implementar redes neurais e realmente estudá-las do ponto de vista da ciência de dados (o que fugiria do escopo do presente trabalho).

No esquema da Figura 1 observamos uma rede neural básica. Cada círculo representa um neurônio. Cada segmento de reta representa uma conexão entre 2 neurônios. Os valores de p representam os pesos das conexões entre neurônios. Os neurônios alinhados verticalmente representam uma camada. Uma rede tipicamente tem uma camada de entrada, uma camada de saída e uma ou mais camadas ocultas. Denotaremos as camadas de entrada com a letra i , as camadas ocultas com a letra h , as camadas de saída com a letra o ⁴. A camada de entrada i é responsável por receber os dados de *input* da rede. Dentro de cada camada os neurônios são numerados de 1 ao número de neurônios daquela camada. As camadas também são numeradas a partir de 1. Chamamos *ativação do neurônio* quando ele recebe um número, seja um número vindo de fora da rede (na camada de entrada) ou quando este número vem das ativações da camada anterior e seus respectivos pesos.

Figura 1 – Rede Neural Básica



Esquema simplificado de rede neural com 2 neurônios na camada de entrada (i), 2 neurônios em uma camada oculta (h) e 2 neurônios na camada de saída (o).

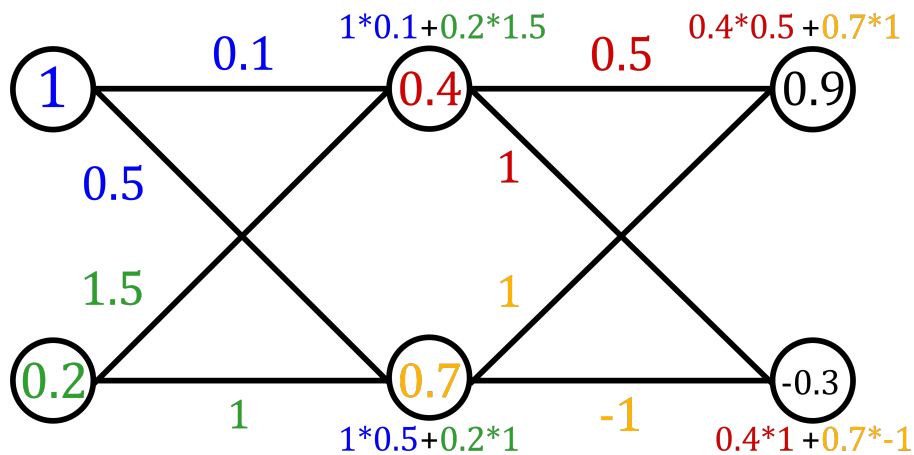
⁴ Do inglês: i de *input*, h de *hidden* e o de *output*.

Cada peso p é representado da seguinte maneira: $p_{m,n}^c$.

- c : índice da camada que será calculada pelos pesos. Ex: a segunda camada na figura 1 é a camada oculta, logo os pesos $p_{1,1}^2$ e $p_{1,2}^2$ serão usados para calcular as ativações da camada h .
- m : índice do neurônio de origem (camada $c - 1$).
- n : índice do neurônio de destino (camada c).

Através das ativações da camada de entrada, as ativações das camadas subsequentes são calculadas, uma a uma, na ordem. Para cada neurônio das camadas intermediárias e de saída, sua ativação é calculada da seguinte maneira: para cada neurônio da camada anterior conectado ao neurônio sendo ativado, multiplica-se a ativação do neurônio da camada anterior pelo peso da conexão entre ambos. Por fim, somam-se todos os produtos. Este processo é exemplificado na Figura (2). Esta propagação unidirecional de valores da camada de entrada até a camada de saída é chamada *Feedforward*.

Figura 2 – Rede Neural Básica, com exemplos de ativação



Rede neural da Figura 1, onde os pesos foram substituídos por valores de exemplo. Cada neurônio na camada oculta e de saída é mostrado junto de sua conta de ativação.

É importante dizer que todo o processo de *Feedforward* pode ser formalmente descrito como uma função matemática cujas variáveis são os números de entrada da camada de *input* e os pesos da rede são constantes (ajustáveis durante o treinamento) que fazem parte desta função.

3.2 Treinando uma rede neural

Uma das maneiras de treinar uma rede neural é uma técnica chamada de *aprendizado supervisionado*. A técnica consiste em utilizar um mapeamento entre os dados de entrada

e suas respectivas saídas esperadas. O *dataset*, portanto, contém os dados de entrada e suas respectivas ativações corretas da camada de saída. Em outras palavras, o *aprendizado supervisionado* consiste em treinar a *IA* usando um conjunto de dados que já contém as “respostas” corretas para cada *input*.

De maneira simplificada o processo de *aprendizado supervisionado* ocorre da seguinte maneira: para cada elemento do conjunto de dados de entrada é calculado o valor das ativações da camada de saída de acordo com os pesos atuais da rede neural. Em seguida esta ativação é comparada com a ativação esperada que existe no *dataset* para calcular o erro. Este cálculo do erro é utilizado para fazer pequenos ajustes nos pesos da rede neural de modo a minimizá-lo. Esta etapa chave do processo de treinamento de uma *IA* é chamada de *backpropagation*⁵. Este processo é feito para todos os elementos do conjunto de dados e, de maneira cíclica, todo o processo se repete um número arbitrário de vezes. Se o processo for bem-sucedido, ao longo das iterações, o erro da rede em calcular as ativações da camada de saída é minimizado. Em essência, é um processo matemático que faz com que a *IA* aprenda a errar menos utilizando cálculo e estatística.

Após o treinamento da rede espera-se que as regras que ela descobriu durante seu treinamento consigam ser *extrapoladas com sucesso para analisar casos externos ao conjunto de dados de treinamento* com a menor taxa de erro possível.

Tomemos um exemplo clássico: treinar uma rede neural para reconhecer dígitos manuscritos. Poder-se-ia usar o banco de dados MNIST de dígitos manuscritos (LECUN; CORTES; BURGESS, 1998) que contém dezenas de milhares de imagens em escala de cinza de dígitos manuscritos retirados do Departamento do Censo dos Estados Unidos e de estudantes do ensino médio. Cada imagem tem a dimensão de 28 pixels de largura por 28 pixels de altura o que dá um total de 784 pixels por imagem.

⁵ *Backpropagation* é uma etapa que envolve competências matemáticas além das esperadas de um músico, que, pelo menos no contexto atual de educação do Brasil, são geralmente ensinadas somente no ensino superior em cursos de ciências exatas.

Figura 3 – Exemplos de imagens contidas no banco MNIT



Visualização de 100 exemplos de imagens retiradas do banco MNIST.

Uma opção seria treinar uma rede com 784 neurônios na camada de entrada: cada um receberia o valor de luminosidade de um dos pixels da imagem. Igualmente tal rede teria 10 neurônios da camada de saída: cada um representando um dos dígitos possíveis que a imagem contém. A escolha do número de camadas ocultas e suas respectivas quantidades de neurônios é arbitrária e tal determinação é competência técnica dos responsáveis pelo treinamento da rede. Na camada de saída o primeiro neurônio pode representar o dígito 1, o segundo neurônio o número 2 e assim por diante até o décimo que representaria o dígito 0. A ativação de cada um desses neurônios representa o nível de confiança da rede neural de que a imagem de entrada contém aquele dígito. Esta representação é geralmente interpretada como uma probabilidade: 1 (ou 100%) seria o nível máximo de confiabilidade enquanto 0 (ou 0%) seria o nível mínimo.

Após um treinamento bem-sucedido, por exemplo, imagine que a rede neural recebeu como *input* uma imagem contendo o número 3. Representando as ativações da camada de saída como um vetor poderíamos ter:

$$\begin{bmatrix} 0.12 & 0.08 & 0.97 & 0.06 & 0.13 & 0.21 & 0.07 & 0.56 & 0.32 & 0.19 \end{bmatrix}$$

O terceiro neurônio, correspondente ao dígito 3, teve a maior ativação: 0.97 (quase 1, que representaria confiança máxima da rede). Assim sendo, o dígito 3 é escolhido como o resultado (*output*) correspondente à imagem de entrada (*input*).

O banco MNIT contém um conjunto de 60,000 imagens para treinamento de redes e um conjunto adicional de 10,000 imagens destinadas a testes para obter uma medição do erro da rede neural em classificar dígitos que não estavam presentes no conjunto de dados de treinamento.

O treinamento desta rede consistiria em, iterativamente, passar as 60.000 imagens de treino para a camada de entrada e comparar o resultado da ativação da camada saída com a resposta correta que está presente no *training set*. O erro do *output* com relação à resposta correta é calculado e ajustes são feitos nos pesos de cada conexão de neurônios a fim de reduzir este erro. Ao fim do processo espera-se que, através dos ajustes no conjunto de pesos da rede neural, esta tenha encontrado uma função matemática que consegue com pouco erro mapear os dados de entrada com suas ativações desejáveis na camada de saída.

Nesta seção nos aventuramos superficialmente em redes neurais e *aprendizado supervisionado*. É importante ressaltar que existem muitas outras formas de redes neurais, de sistemas de *IA* e outros tipos de aprendizado de máquina. Porém, no presente texto não nos aprofundaremos nestes assuntos.

4 “Habilidades” musicais realizadas com Inteligência Artificial

Esta seção descreve diferentes atividades musicais que atualmente podem ser realizadas com Inteligência Artificial. Sua função é proporcionar uma “*visão aérea*” das possibilidades que estas ferramentas proporcionam aos criadores, tanto para situar o leitor no uso da *IA* no mundo da música como também para fornecer uma base para o resto do texto. Não é nosso objetivo fazer um detalhamento extenso ou absolutamente completo de todas as ferramentas e habilidades existentes.

Tentamos categorizar habilidades musicais das *IAs* e cada subseção é dedicada a um tipo de habilidade. Buscamos dentro de cada tipo listar trabalhos que se diferenciem em proposta/metodologia ou outras características de particular interesse. Ou seja, em muitos casos haviam diversas ferramentas/trabalhos que realizavam tarefas musicais de maneiras similares e selecionamos arbitrariamente apenas alguns trabalhos a fim de dar uma visão das diferentes peculiaridades e maneiras de abordar os mesmos objetivos.

As habilidades foram divididas em:

- a) Criação direta;
- b) Derivação;
- c) Acompanhamento *off-line*;
- d) Improvisação;
- e) Mesclagem;
- f) Análise;
- g) Harmonização;
- h) Orquestração;
- i) Expressividade humana;
- j) Reconhecimento ótico de partituras;
- k) Acompanhamento de partitura (*Score Following*);
- l) Hiper-instrumentos.

4.1 Criação direta

Há Inteligências Artificiais capazes de produzir materiais musicais novos sem a necessidade de nenhum dado de entrada (que serviria como “*inspiração*”) por parte do usuário final. A sonoridade do resultado é determinada pelos algoritmos e *training set* utilizados além de eventuais parametrizações disponíveis para o utilizador. Portanto, definimos:

- *Criação direta*: habilidade de criação de ideias musicais sem *input* de material musical pelo usuário final para servir como base.

O projeto *Magenta* da Google oferece bibliotecas homônimas e diversas ferramentas através do *Magenta Studio*⁶. A biblioteca *magenta.js* foi utilizada extensivamente ao longo deste trabalho e é descrita em mais detalhes na Seção 6 (p. 41).

O software *Generate* é capaz de gerar frases e ritmos de 4 compassos. O único parâmetro que o usuário tem para controlar diretamente a qualidade da saída do programa é *temperatura*: o grau de aleatoriedade envolvido no processo. Uma temperatura maior implica em maior aleatoriedade.

O software usa o modelo *Music VAE* que pode ser utilizado programaticamente através das bibliotecas da *Magenta*, como a *magenta.js* (ROBERTS; HAWTHORNE; SIMON, 2018). As bibliotecas oferecem um nível maior de controle sobre o modelo se comparadas ao *Magenta Studio*. O time do projeto disponibiliza **modelos pré-treinados**⁷, mas o usuário também pode treinar o modelo com seu próprio *dataset*.

A biblioteca ainda disponibiliza o modelo *MidiMe*, que é uma versão do *Music VAE* com o diferencial de que pode ser facilmente treinado com um dataset limitado de um único exemplo para gerar ideias parecidas com o material original.

Ainda nesta categoria de habilidades também consideramos *IAs* que conseguem compor no estilo de um ou mais compositores. O exemplo canônico de *IA* capaz de “emular” o estilo de outros compositores é a EMI de David Cope (também conhecida pelo apelido *Emmy*):

Certainly the best-known work on computer composition using AI is David Cope’s EMI project (Cope, 1987, 1990). This work focuses on the emulation of styles of various composers. It has successfully composed music in the styles of Cope, Mozart, Palestrina, Albinoni, Brahms, Debussy, Bach, Rachmaninoff, Chopin, Stravinsky, and Bartok. (MÁN-TARAS, 2016, p. 7)

Tal habilidade pode parecer diferente das habilidades fornecidas pela *Magenta*, por exemplo, mas é apenas um caso específico onde o *training set* e o algoritmo são escolhidos com o intuito de replicar a gramática de um ou mais compositores específicos. Ao treinar os modelos da *Magenta* nas obras de compositores famosos provavelmente obter-se-iam resultados similares, porém seus modelos foram treinados em *datasets* mais genéricos.

Em outras palavras, alcançam-se resultados diferentes por ter objetivos diferentes, porém, sob certo ponto de vista, as técnicas são similares.

Por tal motivo subdividimos a habilidade de **criação direta** em:

- a) *emuladora*: quando os algoritmos e *training sets* são escolhidos visando imitar o estilo de um ou mais compositores;

⁶ Disponível como uma suíte de aplicativos autônomos ou como *plugins* para o software *Ableton Live*.

⁷ Chamados de *checkpoints* de modelos. São configurações salvas dos pesos da rede neural após treinamento.

- b) *genérica*: quando os algoritmos e *training sets* são escolhidos com o objetivo de fornecer ferramentas para serem usadas em diversos contextos e estilos.

A criação direta pode ser utilizada para criar frases, trechos, excertos, melodias, ideias, etc. Porém, também podem ser criadas para criar composições inteiras. Há soluções no mercado que conseguem criar composições inteiras com poucos cliques e em diversos estilos musicais conhecidos (AIVA:..., 2021).

4.2 Derivação

Nesta seção abordamos Inteligências Artificiais capazes de criar derivações a partir de materiais musicais passados como dados de entrada. Estes materiais de entrada são utilizados para influenciar de maneira direta o resultado da tarefa realizada pelo software e são geralmente criados ou selecionados pelo usuário, ou seja, determinados (pelo usuário) numa etapa posterior ao desenvolvimento e treinamento da *IA*.

A *magenta.js* permite criar variações (no sentido de derivações que soam similares) diretas de ideias de *input* através de seu modelo *MusicVAE*. Novamente é utilizado um parâmetro de *temperatura* para controlar o nível de aleatoriedade envolvido no processo. Há um segundo parâmetro que é o grau de similaridade, especificado de zero a um, sendo 0 o menos similar e 1 o mais similar.⁸

Outro tipo de derivação é criar continuações das ideias musicais de entrada. O resultado não precisa ser tão similar quanto uma variação, mas precisa, idealmente, fazer sentido quando executado logo após a ideia fornecida. O *Magenta Studio* fornece o software *Continue* que é capaz de estender/continuar uma sequência de notas de entrada em até 32 compassos. O modelo por trás da ferramenta é o *MusicRNN* que também está disponível na *magenta.js*.

4.3 Acompanhamento *off-line*

Há softwares de *IA* que conseguem gerar acompanhamentos para ideias musicais fornecidas como dados de entrada.

Este tipo de acompanhamento é realizado “*Off-line*” como no caso da ferramenta *Drumify* do Magenta Studio, que aceita como entrada um arquivo MIDI qualquer e de saída devolve a música original acrescida de um acompanhamento percussivo.

4.4 Improvisação

As *IAs* também podem gerar acompanhamentos em tempo-real. Diferenciamos esta habilidade da habilidade “*Acompanhamento off-line*”, pois, devido à sua natureza *live* tais

⁸ Atualmente ainda não há tal funcionalidade no *Magenta Studio*.

softwares podem apresentar capacidades bastante sofisticadas de interação, criando diálogos em tempo real com músicos no palco ou no estúdio, permitindo um ciclo constante de *input* e *output* na ferramenta. Um exemplo de tal *software* é o *Reflexive Looper* do projeto Flow Machines (PACHET; ROY; CARRÉ, 2021, p. 22).

Pode-se arguir que a [Criação direta](#) pode ser realizada em tempo-real, porém não possuiria tais características de responsividade.

Similar aos casos de [Criação direta](#), sistemas improvisadores podem ser *genéricos*, como o sistema *GenJam*, ou *emuladores*, como o sistema de Judy A. Franklin, capaz de improvisar no estilo do saxofonista *Sonny Rollins* (MÁNTARAS, 2016, p. 114).

Estes são exemplos de improvisos cuja interação é feita entre *humano e IA* mas também há sistemas capazes de modelar as interações inter-músicos que ocorrem durante uma performance, permitindo que diferentes agentes de *IA* consigam tocar juntos e se comunicarem durante a performance, ou seja, um modelo de improviso *IA - IA* (PUERTO; THUE, 2017).

4.5 Mesclagem

Esta seção lista as Inteligências Artificiais capazes de criar algum tipo de combinação a partir da mistura de duas ou mais ideias ou dados musicais. Tais habilidades podem utilizar dados fornecidos diretamente por humanos ou o *output* de alguma ferramenta de *IA* além de combinações entre ambos.

A *Magenta Studio* fornece *Interpolate*, software que é capaz de gerar misturas, em diferentes proporções, de 2 melodias ou de 2 ritmos. A biblioteca *magenta.js* também fornece tal funcionalidade e a expande para também conseguir fazer misturas de 4 ideias diferentes.

O *NSynth*, também do time da Magenta, é um sintetizador capaz de fazer mistura de timbres entre 2 ou 4 instrumentos, também em diferentes porcentagens (NSYNTH:..., 2021). Sons inusitados podem ser utilizados como instrumentos (por exemplo: pode-se ouvir em seu website o som de um gato misturado com o de uma flauta).

4.6 Análise

Existe uma área de pesquisa chamada Recuperação de Informação Musical, ou *MIR*, do inglês *Musical Information Retrieval*, que aborda o problema de extrair diversos tipos de informação de dados musicais.

Bharucha (1993) descreve o modelo MUSACT que é capaz de analisar a harmonia de uma peça. A análise observa as “intuições” da música: o aspecto de expectativa e resolução (ou não-resolução) que a harmonia gera. Uma das vantagens do modelo é que ele funciona com relações gradativas e ambíguas:

One of the advantages of the model is that it enables chord and key instantiations to be graded and ambiguous. These ambiguities are an important feature of music, exploited during modulations or other transitions, and used to create graded degrees of expectancy violation. (BHARUCHA, 1993, p. 499)

Apesar das observações a cerca de seu uso em música atonal (BHARUCHA, 1993, p. 501) o modelo trabalha apenas com tríades maiores e menores (a tríade diminuta teria a mesma função de uma tríade maior de fundamental uma terça maior abaixo). O modelo consegue capturar os princípios mais fundamentais da harmonia tonal, por exemplo: ao receber como dados de entrada os acordes de Fá maior e Sol maior, corretamente dá a Dó maior a maior pontuação dentre todas as tonalidades, mesmo o acorde de Dó maior estando ausente no *input*. O modelo apresentou resultados similares a humanos em testes de percepção. O MUSACT consegue levar o contexto em consideração: as progressões Dó Maior - Fá Maior e Fá maior - Dó maior têm ativações diferentes quando o modelo acha que está em Dó maior versus quando acha que está em Fá maior. Notoriamente, o círculo de quintas surge naturalmente nas ativações do modelo. Uma sequência de acordes que sugere Sol Maior resultaria numa grande ativação da unidade de Sol maior, seguida de uma ativação menos expressiva em Dó Maior e Ré Menor e depois uma ativação ainda mais baixa de Fá Maior e Lá maior, etc.

4.7 Harmonização

Huang et al. (2017) descreve um modelo, COCONET, capaz de gerar harmonizações bastante convincentes. A ferramenta foi utilizada para criar um *Doodle do Google para o Aniversário do Bach*⁹. Para a realização do *Doodle*, COCONET recebeu treinamento a partir de 306 composições de Bach. As composições eram apresentadas com notas faltantes, aleatoriamente retiradas da composição original, e a IA fazia uma previsão de quais eram as notas que foram retiradas. Por fim a rede neural fazia uma comparação entre sua previsão e a música original e, iterativamente, através deste processo, conseguia se tornar melhor em fazer tais previsões (mais um exemplo de *aprendizado supervisionado*).

Assim sendo, para realizar uma harmonização de uma ideia de *input*, basta tratá-la como um coral de Bach com notas faltantes e preencher as lacunas.

4.8 Orquestração

Utilizando ferramentas criadas no projeto *Flow Machines* foram criadas diversas orquestrações da obra “Ode à Alegria” de Beethoven (PACHET; ROY; CARRE, 2021, pg. 9). As orquestrações foram produzidas com base em estilos extraídos de:

⁹ *Doodles* são versões customizadas e temporárias do logo da Google em sua *homepage*. Em alguns casos é um mini aplicativo.

- a) Transcrições para MIDI¹⁰ dos corais de Bach;
- b) “Violão Brasileiro”/Bossa Nova;
- c) Grupo vocal *Take 6*;
- d) “Transcrições MIDI” de lounge music "Best of Café del Mar";
- e) A gravação de *Penny Lane* dos *The Beatles*;
- f) A gravação de *Chi Mai* de *Ennio Morricone*;
- g) A gravação de *Prayer in C* de *Lilly Wood and the Prick*.

4.9 Expressividade humana

Há ferramentas com diversos fins que têm a habilidade de simular as imperfeições humanas numa performance musical. Estas expressividades se manifestam na forma de variações temporais de ritmo, articulação, intensidade, andamento, etc.

Algumas têm como objetivo adicionar *expressividade humana* em um material musical dado (particularmente útil para material gerado por computadores). O software *Groove*, criado com base no modelo *GrooVAE*, da *Magenta Studio* é capaz de adicionar tais expressividades em um arquivo MIDI de bateria de entrada (arquivo este que tipicamente foi gerado via computador). (GILLICK et al., 2019)

Teoricamente tal processo até poderia ser realizado manualmente por um humano ao tratar o arquivo MIDI em um software *DAW*, por exemplo. Porém, tal trabalho consumiria uma quantidade incomensurável de tempo e seria extremamente repetitivo e entediante uma vez que ele consiste em tomar uma enorme série de micro-decisões musicais sobre *timing* e/ou articulação, intensidade, etc nota por nota porém, também levando em consideração todas as estruturas superiores como compasso, frase, etc.

Algumas ferramentas possuem mecanismos internos para lidar com estas expressividades enquanto realizam outras tarefas como o sistema *SaxEx*, capaz de criar performances expressivas de saxofone, e um projeto derivado *TempoExpress*, capaz de fazer mudanças de andamento/BPM (*tempo*) enquanto garante que estas micro relações temporais de expressividade se mantenham adequadas (MÁNTARAS, 2016, p. 110).

4.10 Reconhecimento ótico de partituras

O reconhecimento ótico de partituras ou OMR do inglês *Optical Musical Recognition* é o equivalente musical do *OCR*¹¹ para partituras. O software pode, por exemplo, receber uma imagem de partitura escaneada e convertê-la em um formato digital como uma

¹⁰ Não se sabe exatamente o que “MIDI” significa neste contexto uma vez que é um protocolo de troca de informações musicais e não um instrumento.

¹¹ *Optical Character Recognition*: tecnologia que permite extrair texto de imagens.

sequência *MIDI*, *Music XML* ou o formato de algum editor de partituras (PACHA; EIDENBERGER, 2017).

Discutivelmente tal habilidade pode parecer uma habilidade visual e não musical, porém a entendemos como uma capacidade que tem efeitos práticos e impactos diretos na área da música e portanto tem sua relevância.

4.11 Acompanhamento de partitura (*Score Following*)

Uma *IA* que mostra esta habilidade consegue “ouvir” um áudio enquanto acompanha uma partitura e determinar em que ponto da partitura o áudio está. Esta habilidade pode ainda ser complementada com subseqüentes ações tomadas pela parte do computador para, por exemplo, disparar eventos musicais em pontos pré-determinados de uma composição de maneira adaptativa às nuances da performance de música humanos.

Dorfer, Arzt e Widmer (2016) descrevem um sistema capaz de relacionar o instante atual de um áudio com sua partitura e apresenta uma característica interessante: isto funciona com imagens de partituras, ou seja, sem a necessidade de a música estar representada em um formato específico para informações musicais (como um arquivo MIDI). Tal sistema ainda é muito limitado perto de competidores como *Antescofo* (CONT, 2008), porém promissor e não necessita de uma partitura aumentada (*augmented score*).

4.12 Hiper-instrumentos

Sistemas de Inteligência Artificial podem ser integrados em tempo real através de diversos sensores a instrumentistas e seus instrumentos musicais, utilizando um conjunto de diversas habilidades musicais diferentes para criar ainda um novo instrumento, um *hiper-instrumento*, resultado da interação de todas estas partes.

Notoriamente *Tod Machover* criou um hiper-violoncelo e o utilizou em sua obra *Begin Again Again...*, que foi tocado pelo renomado violoncelista Yo-Yo Ma em seu *premiere* no *Tanglewood Festival*. (MÁNTARAS, 2016, p. 112)

5 Jogo: *Carmim*

Nesta seção vamos descrever o processo de design e implementação do jogo de cartas de composição musical *Carmim*: **C**artas **M**usicais **I**nteligentes feitas com **M**agenta. Primeiramente pode-se perguntar: por que um jogo? O que um jogo pode trazer de diferencial quando comparado a um software de criação tradicional? Respondemos:

- a) Jogos geralmente impõe desafios a serem superados. Estes desafios muitas vezes aparecem na forma de limitações. Estas limitações podem inclusive amplificar a criatividade (JOYCE, 2009). Um jogo é, essencialmente, um sistema (SALEN; ZIMMERMAN, 2012). Um jogo de criação é, portanto, praticamente um convite a usá-lo criativamente; ser criativo ao ser criativo!;
- b) Além de suas possibilidades adicionais, um jogo pode englobar todas as possibilidades de sistemas tradicionais ao incorporar um modo de criação livre¹²;
- c) Jogos são divertidos! Jogos podem ser um fim por si só, assim como a música.

Escolhemos fazer o jogo como um aplicativo web, ou seja, que roda no navegador, visto que isto facilita seu uso: não é necessário nenhum tipo de instalação, basta acessar sua URL. Além disso, funcionará em múltiplas plataformas (Windows, Linux, OSX).

O jogo é de *código-livre* e seu repositório pode ser encontrado na página do GitHub do autor: <https://github.com/HenriAugusto/TCC>. O repositório contém tanto o código como uma versão *live* do jogo na URL: <https://henriaugusto.github.io/TCC/>.

Buscando dentre as ferramentas disponíveis uma que se adequasse à nossa finalidade, encontramos e escolhemos a biblioteca *open-source magenta.js*. Esta biblioteca permite utilizar as ferramentas desenvolvidas no projeto *Magenta*¹³ diretamente no navegador.

Como princípio de design resolvemos fornecer ao jogador, através de um sistema de cartas, acesso às operações fundamentais da biblioteca *magenta.js*.

¹² Exemplos de modos de criação livre são abundantes em jogos que apresentam algum aspecto criativo: *Rollercoaster Tycoon 2* é um jogo de simulação em que você assume o papel de um magnata dono de parques de diversões e contém um modo de construção livre de montanhas russas; *Subnautica* é um jogo de sobrevivência com história e ambiente imersivos, porém tem um modo de construção livre de bases subaquáticas; *Minecraft* é extremamente popular principalmente devido ao seu *Creative Mode* que permite construir ilimitadamente; fora uma infinidade de jogos, dos mais diversos gêneros, que contém um editor de níveis. Alguns jogos são inclusive apenas *sandboxes*, universos onde você pode construir e experimentar à vontade com um sistema, como, por exemplo: *Garry's Mod*

¹³ Um projeto maior criado pelo *Google Brain Team* para explorar IA e artes.

5.1 Visão geral da interface

Nesta seção descreveremos os principais elementos da interface do jogo e suas funções. Na Figura 4 cada elemento recebeu uma numeração que o relaciona com as descrições:

Figura 4 – Interface do jogo



Screenshot da interface do jogo com elementos da interface numerados.

1. *Input box* onde o jogador escreve o nome da peça/sessão/partida;
2. **Timeline**: cada linha laranja é uma pista (ou *track*) onde o jogador consegue soltar as cartas que contém sequências musicais para criar a composição;
3. Cada retângulo no arranjador é uma sequência musical que veio de uma carta. Ao clicar no retângulo a carta retorna à “mão” do jogador;
4. **“Mão” do jogador**: É onde ficam as cartas disponíveis para o jogador usar;
5. Menu que permite salvar e carregar a peça/sessão/partida.;
6. Uma única carta;
7. **Baralho**: o jogador pode clicá-lo para sacar cartas;
8. Uma “régua” onde o jogador pode clicar para iniciar o playback daquele ponto da composição;
9. Botões de *Play* e *Stop* para controle básico de *playback* da música;

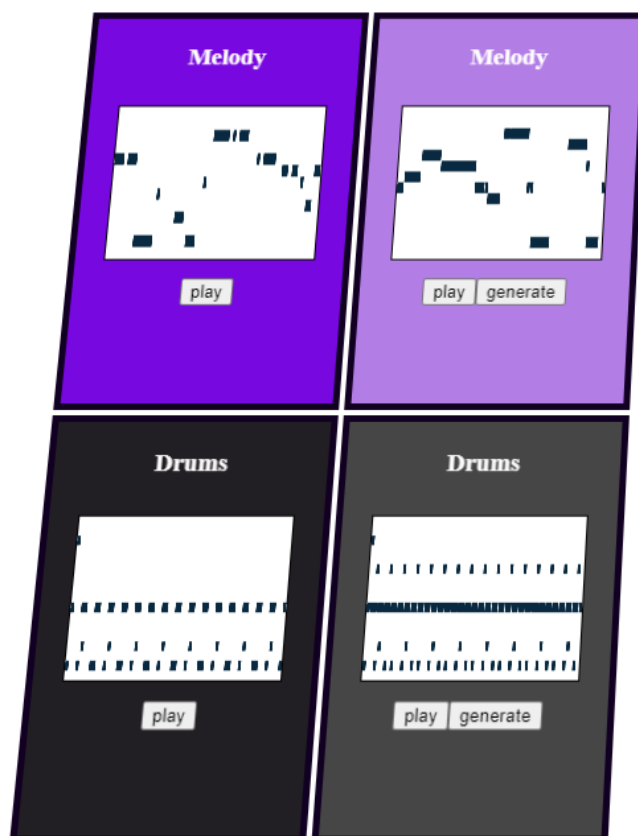
10. Dois *sliders* para controlar a *temperatura* dos dois modelos: MusicVAE e MusicRNN¹⁴. Em termos gerais, estes sliders controlam o nível de aleatoriedade implicado nas operações de *IA*.

5.2 As cartas

Esta seção descreve em detalhes os tipos de cartas que fazem parte do jogo.

5.2.1 Cartas de sequência: melodia, bateria, geradoras e não-geradoras

Figura 5 – Cartas de sequência



Canto superior esquerdo: carta de melodia fixa. Canto superior direito: carta de melodia geradora. Canto inferior esquerdo: carta de bateria fixa. Canto inferior direito: carta de bateria geradora.

As cartas de sequência são as unidades mais básicas do jogo. Cada carta de sequência pode ser arrastada para as pistas da *Timeline* para criar a composição. Existem quatro tipos de cartas de sequência:

¹⁴ A descrição destes modelos será feita a seguir.

1. Carta de melodia fixa;
2. Carta de melodia geradora;
3. Carta de bateria fixa;
4. Carta de bateria geradora.

Na Figura 5 podemos ver um exemplo de cada um dos quatro tipos. As cartas de melodia podem apenas conter melodias e as cartas de bateria apenas ritmos. Não é possível converter uma na outra. As cartas geradoras se diferem das fixas por um botão adicional que permite gerar uma nova melodia ou ritmo para a carta (sobrescrevendo a ideia anterior).

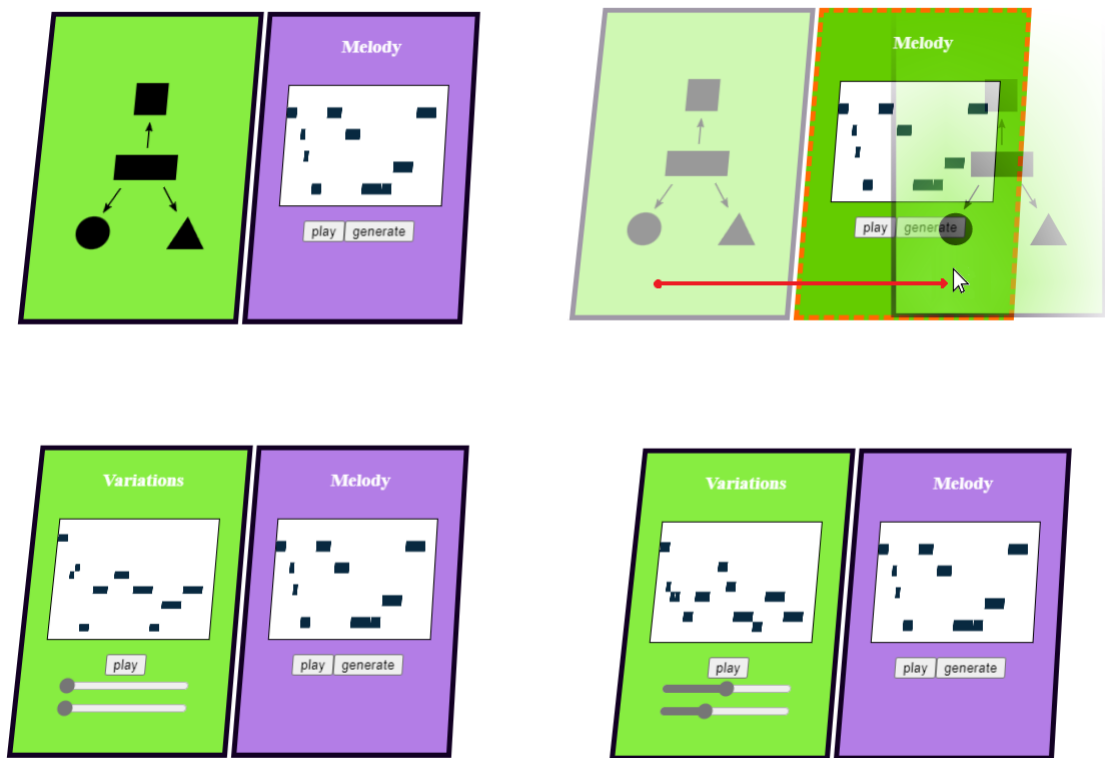
5.2.2 Carta de variação

A *carta de variação* permite ao jogador criar diferentes variações em diferentes níveis de similaridade a partir de uma ideia original. A carta, ao ser arrastada para outra carta de sequência, cria uma série de variações a partir da sequência da carta alvo. Este processo é ilustrado na Figura 6 e funciona tanto para criar variações de sequências de melodia quanto de bateria.

A carta contém dois *sliders*: o primeiro serve para controlar o nível de proximidade da variação, ou seja, se as variações serão mais próximas ou mais distantes da ideia original. O segundo *slider* alterna entre as diversas variações geradas para o nível de similaridade selecionado.

Como a carta de variação, após ativada, contém sequências musicais, ela pode ser tratada como uma carta de sequência! Isto é, pode-se arrastar uma carta de variação sobre outra carta de variação para criar variações a partir de uma variação. O mesmo pode ser dito para as outras operações realizadas com cartas de sequências como veremos a seguir, ou seja, para efeito de operações de *IA* esta carta pode ser tratada como se fosse uma carta de sequência.

Figura 6 – Carta de variação



No canto superior esquerdo temos uma carta de variação e uma carta de sequência melódica. No canto superior direito ilustramos a carta de variação sendo arrastada até a carta de sequência melódica. Nos cantos inferior esquerdo e inferior direito podemos ver a carta de variação após ter sido arrastada até a carta de melodia, com diferentes configurações.

5.2.3 Carta de continuação

A *carta de continuação* é utilizada para gerar novas sequências que soam como a continuação da ideia alvo. Ela deve ser arrastada até a carta de sequência que contém a sequência alvo e a partir deste ponto poderá gerar uma infinidade de continuações diferentes.

Ao ser arrastada para uma carta de sequência, inicialmente ela gerará uma única continuação para a sequência da carta alvo, porém o jogador pode solicitar quantas continuações quiser ao clicar em um botão da carta. Ilustramos a carta de continuação na Figura 7.

Figura 7 – Carta de continuação



No canto superior esquerdo temos uma carta de continuação e uma carta de sequência melódica. No canto superior direito ilustramos a carta de continuação sendo arrastada até a carta de sequência melódica. Nos cantos inferior esquerdo e inferior direito podemos ver a carta de continuação após ter sido arrastada até a carta de sequência, com duas versões diferentes de continuações.

O jogador poderá trocar a sequência alvo da carta de continuação a qualquer momento. Basta arrastá-la para a carta contendo a nova sequência alvo.

Assim como a carta de variação, a carta de continuação funciona com sequências de melodia e de bateria e também pode ser tratada como uma carta de sequência!

5.2.4 Carta de interpolação

A *carta de interpolação* é utilizada para misturar duas sequências. Ela é obtida ao arrastar uma carta de sequência sobre outra carta de sequência. Como consequência uma terceira carta é criada, uma carta de interpolação, que contém sequências que são misturas das sequências originais em diferentes proporções. Este processo é ilustrado na Figura 8.

Figura 8 – Carta de interpolação



Ao arrastar uma carta de sequência para outra é gerada uma carta de interpolação. Este tipo de carta contém um *slider* que controla a interpolação. Quando o valor do *slider* está em uma das extremidades, a carta contém uma das sequências originais. Para os outros valores a carta conterá uma mistura entre as duas ideias.

Na versão atual do jogo a interpolação é feita em 16 etapas, sendo as etapas 1 e 16 as duas sequências originais a serem misturadas. As etapas intermediárias representam misturas das sequências originais em diferentes medidas. O usuário escolhe qual quer usar por meio de um *slider* na carta. Como a carta de interpolação também contém sequências, também pode ser tratada como carta de sequência para ser alvo de operações que afetam este tipo de carta.

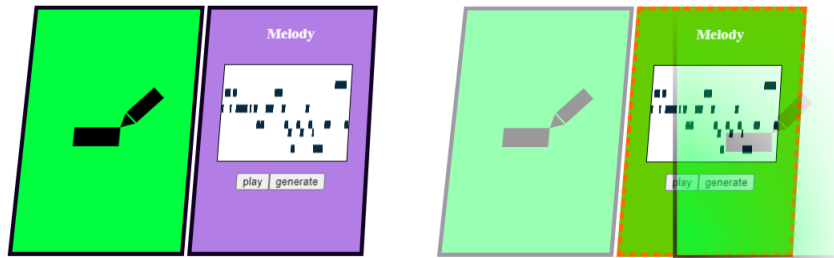
5.2.5 Carta de edição

A carta de edição é utilizada para editar ou, possivelmente, recriar¹⁵ uma sequência. Para tal basta arrastá-la sobre uma carta de sequência. A interface do editor (Figura 10) será aberta e o jogador poderá editar cada nota da sequência além de deletá-las e acrescentar

¹⁵ Basta apagar todas as notas presentes e criar algo do zero.

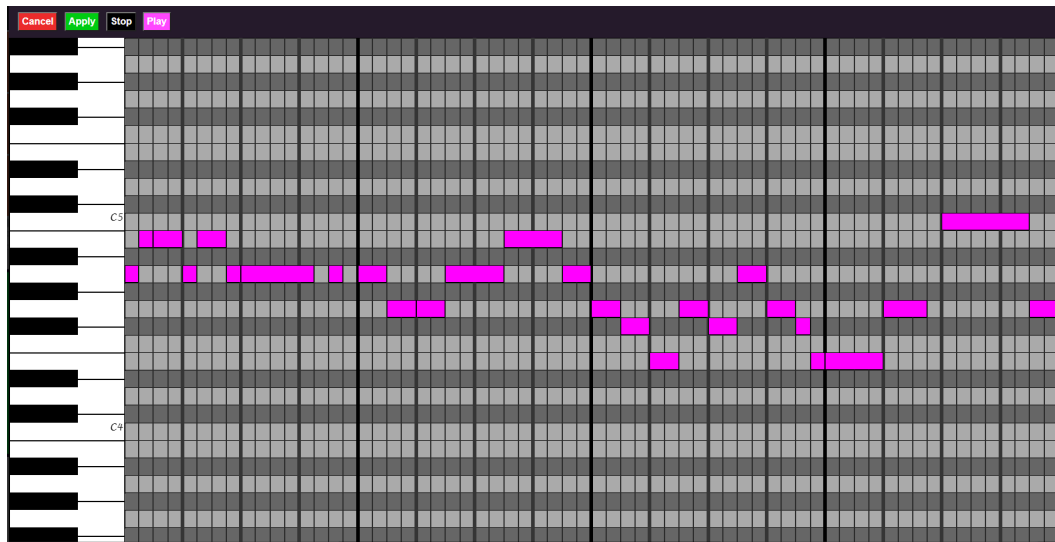
outras. Como referência, há teclas de piano na parte esquerda do editor. Estas teclas são iluminadas dependendo da linha sobre a qual o mouse se encontra para facilitar a edição. Pode-se também clicar sobre as teclas para ouvir sua nota correspondente¹⁶. No editor notas podem ser acrescentadas, removidas e movidas horizontalmente (temporalmente) ou verticalmente (transposição).

Figura 9 – Carta de edição



A carta de edição pode ser arrastada para cima de uma carta de sequência. Esta ação abrirá um editor que permite ao jogador editar ou recriar a sequência.

Figura 10 – *Piano Roll*: interface da carta de edição



Exemplo de uma sequência sendo carregada para edição no editor. A interface do editor é do tipo *rolo de pianola*, ou *piano roll*, em inglês.

¹⁶ A mesma interface é utilizada para sequências de bateria. Neste caso cada som de bateria será mapeado para uma tecla do piano.

5.3 Playtests

A realização de *Playtestings* é parte fundamental do *design* de jogos bem-sucedido:

[...] a interação de um jogo sempre surpreende seus criadores, especialmente se o *design* do jogo for incomum ou experimental. Mesmo um *designer* veterano não pode prever exatamente o que vai e o que não vai funcionar antes de experimentar o jogo em primeira mão. Faça o protótipo do seu jogo antecipadamente. Jogue-o por todo o processo de *design*. Recrute o maior número possível de pessoas para jogar o seu jogo e observe-as jogando. Permita que você seja surpreendido e desafiado. Seja sempre flexível. E não esqueça de se divertir. (SALEN; ZIMMERMAN, 2012)

Realizamos um processo iterativo de criação de jogo. Como ponto de partida optamos por criar um sistema que fornecia a ferramenta que parecia ser a mais básica de todas: a geração de melodias e ritmos. Ou seja, as cartas que posteriormente vieram a ser chamadas de *cartas de sequência fixas*.

Cada sessão de *playtesting* consistiu em uma ou mais partidas, porém todas as partidas da mesma sessão foram realizadas na mesma versão do jogo. A seguir, há as anotações que fiz após cada uma das 9 sessões realizadas. Em cada registro incluo meus pensamentos e análise sobre a sessão atual assim como as ideias que tive para futuras modificações no jogo a fim de direcionar a continuidade de seu desenvolvimento. Em retrospectiva, analisaremos este processo para refletir sobre a experiência em sua totalidade e sobre as habilidades da *magenta.js* que foram implementadas.

5.3.1 Playtest 1

Neste estado, apenas com *cartas de sequência fixas*, o jogo é simplesmente muito frustrante, pois muitas das ideias geradas pela *IA* não me soavam muito interessantes. O jogador acaba ficando com uma mão cheia de cartas ruins, sem vontade de usá-las. Além disso, as ideias eram completamente desconexas entre si.

Futuro desenvolvimento: criar cartas de sequência que permitam solicitar novas melodias para a *IA* para mitigar o problema de apenas ter ideias que não são interessantes para se trabalhar.

5.3.2 Playtest 2

A implementação de cartas de sequência que permitem requisitar novas melodias/ritmos (*cartas de sequência geradoras*) trouxe uma melhoria perceptiva, porém ainda assim é difícil juntar ideias de maneira coerente, porque muitas ideias geradas ainda são bastante diferentes entre si. Além disso, o processo de gerar melodias até que uma delas faça sentido com as antigas é extremamente repetitivo e demorado.

Futuro desenvolvimento: implementar uma carta que proporcione alguma das outras funcionalidades da biblioteca que poderia ajudar a criar ideias mais coerentes entre si.

5.3.3 Playtest 3

De modo a tentar permitir gerar ideias que “conversem” mais, a funcionalidade escolhida para ser implementada foi a de interpolação. Implementamos a *carta de interpolação* que permite misturar duas melodias. A melhoria foi mais significativa do que a *implementação de cartas de sequência geradoras*, pois agora é um pouco mais fácil de gerar ideias relacionadas entre si de maneira coerente. Ainda assim ao jogar fica uma impressão que as possibilidades continuam muito limitadas.

A carta de interpolação foi implementada da seguinte maneira: ao se arrastar uma primeira carta de sequência até uma segunda carta de sequência é criada uma terceira carta, a carta de interpolação. Isto teve uma consequência imprevista: a carta de interpolação é uma maneira de duplicar uma carta de sequência qualquer. Basta gerar uma carta de interpolação com a carta de sequência que se deseja duplicar e uma outra carta qualquer. Em seguida basta manter o *slider* de interpolação completamente à esquerda ou à direita de forma a utilizar a versão original da carta a ser duplicada.

Notei uma certa angústia por não poder corrigir uma ou duas notas que estavam “fora do lugar” em melodias que de outro modo seriam perfeitamente interessantes.

Futuro desenvolvimento: criar uma carta de edição que permite ao jogador editar as sequências de uma carta. Isto pode ajudar o jogador a lidar com estas notas fora da harmonia.

5.3.4 Playtest 4

Implementamos a carta que ficou conhecida como *carta de edição*. Durante este *playtest*, para testá-la, fiz com que ela sempre aparecesse na mão do jogador no início da partida. Adicionalmente ela poderia ser usada quantas vezes o jogador desejasse.

Reparei que, estando sempre disponível, esta carta era usada com muita frequência. Inclusive poderia ser utilizada para completamente sobrescrever a sequência de uma carta de sequência, quase como uma carta de trapaça. Isto não parecia necessariamente ser ruim, porém imaginamos que a chance de sacar esta carta deve ser reduzida e/ou ela deve ser utilizável apenas uma vez.

Futuro desenvolvimento: continuar a implementação de outras funções básicas da biblioteca.

5.3.5 Playtest 5

Esta sessão foi conduzida após a implementação da *carta de continuação*. Nesta versão a carta de continuação gerava apenas uma sequência, ou seja, não era possível requisitar outras continuações uma vez que a carta é usada.

Além disso, uma limitação do modelo de *IA* acabou se tornando aparente: os *checkpoints*¹⁷ utilizados para gerar/processar as cartas de melodia apenas são capazes de gerar material monofônico. Entretanto, a carta de edição permite gerar materiais polifônicos em uma única carta. Isso implica que, por exemplo, ao tentar gerar a continuação de uma ideia polifônica (criada via carta de edição), a continuação seria apenas monofônica. Esta é uma limitação da biblioteca. Há um *checkpoint* que pode gerar ideias polifônicas, o *checkpoint* de trios, porém este apresenta um problema para nossos fins: foi feito para gerar três ideias de uma vez (uma melodia, uma linha de baixo e percussão), ou seja, uma configuração limitada à *exatamente* 2 linhas melódicas monofônicas e uma percussiva.

Em outra nota: o jogo perde um pouco a noção de progresso gradual uma vez que o jogador pode simplesmente sacar todas as cartas do baralho a qualquer momento. Isso também faz com que seja muito confuso encontrar as cartas desejadas. Além disso, ter menos informações com as quais trabalhar pode potencializar a criatividade. (JOYCE, 2009)

Futuro desenvolvimento: o jogo agora parece dar funcionalidades suficientes para conseguir criar uma ideia completa, de uma quantidade razoável de compassos (entre 8 e 16). Começa a se tornar necessário um mecanismo de salvar e carregar a partida. Para o próximo passo será importante programar um sistema de geração de cartas onde se pode controlar a probabilidade de cada carta aparecer. Também resta testar a implementação de um limite de cartas na mão no jogador.

5.3.6 Playtest 6

Um sistema que permite salvar/carregar a partida foi implementado. Durante estas sessões a maior probabilidade era receber *cartas de sequência fixas*. Estas cartas se provaram mais um desafio do ponto de vista de um jogo, uma vez que muitas delas contêm ideias com as quais é difícil de se trabalhar e, por não serem utilizadas na composição, acumulavam-se na mão do jogador atrapalhando a fluidez do jogo. Algumas soavam demasiadamente aleatórias, sem direcionalidade, e por vezes com tessituras extremamente amplas e texturas exageradamente pontilhistas. Paralelamente, implementamos um limite de 10 cartas que podem ser seguradas por vez na mão do jogador.

Futuro desenvolvimento: Diminuir a probabilidade de *cartas de sequência fixas* e aumentar a probabilidade de *cartas de sequência geradoras*. Similarmente como fizemos ao permitir ao jogador requisitar várias sequências através das cartas geradoras, é interessante

¹⁷ Um *checkpoint* é um conjunto pré-definido de valores para os pesos da rede neural. Estes valores são obtidos através do treinamento da *IA* e armazenados para uso posterior.

permitir requisitar infinitas continuações através da carta de continuação. Isto poderia mitigar casos em que o resultado inicial é de pouco valor musical. O limite de 10 cartas se provou muito severo: aumentar o limite de cartas na mão do jogador. Finalmente, implementar a funcionalidade de *variação* da biblioteca pode deixar o jogo mais fluído. Na versão acaba-se usando a carta de interpolação para gerar variações (por exemplo, para variar a carta de sequência X : interpolava X com uma sequência Y qualquer e fazia uma mistura, por exemplo, 80% de X e 20% de Y)

5.3.7 Playtest 7

A *carta de variação* foi implementada. Com esta adição consegui fazer os primeiros protótipos que pareciam promissores e com isto os *playtests* tinham *timelines* cada vez mais preenchidas. Porém, ainda não havia nenhuma maneira de iniciar o *playback* do meio da música. Assim sendo, para fazer experimentos com ideias no final da linha do tempo era necessário ouvir a composição desde o início o que tornava o processo moroso e entediante.

Futuro desenvolvimento: implementar uma maneira de iniciar o *playback* de qualquer ponto da música.

5.3.8 Playtest 8

Esta seção de playtesting foi bastante longa e interessante. Poder iniciar o *playback* de qualquer ponto da música se provou uma funcionalidade básica indispensável que tornou as partidas muito mais fluídas. O acréscimo da *carta de variação* teve um efeito similar na fluidez do jogo. Esta melhoria permitiu que outros pontos problemáticos ficassem evidentes: a situação mais comum no jogo é ficar com uma mão cheia de cartas de sequências, com várias sequências que não se encaixam na composição. Enquanto isso, reparei que frequentemente trabalhava sempre próximo do limite da mão do jogador (o que era um empecilho). Me encontrava usando principalmente as cartas de continuação, variação e interpolação. Isto não aparenta ser coincidência, uma vez que variações e derivações são procedimentos básicos da composição musical e estas semelhanças entre ideias são um dispositivo de memória que gera coerência no discurso musical.

Além disso, o limite de três pistas na *linha do tempo* se tornou indesejável.

Futuro desenvolvimento: aumentar a probabilidade das cartas de continuação e variação e diminuir as probabilidades de cartas de sequência fixas e geradoras, isto é, aumentar a quantidade de cartas que permitem derivar ideias a partir dos materiais musicais escolhidos pelo jogador e diminuir a quantidade de cartas com ideias novas. Aumentar o número de pistas na *timeline* para cinco.

5.3.9 Playtest 9

Os ajustes nas probabilidades das cartas para enfatizar as cartas de derivação deixou o jogo mais divertido e trouxe uma sensação maior de liberdade para experimentar. Nesta versão completamos o primeira excerto musical relativamente longo no *Carmim* que sentimos ser genuinamente satisfatório para, se não utilizado literalmente, pelo menos ser utilizado como base para o desenvolvimento de uma composição completa (Ver apêndice “[Excerto criado com o jogo Carmim](#)”). Também nesta sessão ficamos finalmente convencidos de que seria interessante se houvesse uma ferramenta que fosse capaz de arranjar diferentes ideias musicais de maneira que soassem bem juntas, seja por vias contrapontísticas, acórdicas ou outras. Mesmo com mais liberdade para fazer derivação de ideias, continua sendo muito comum necessitar fazer pequenos ajustes em algumas notas para ter um resultado satisfatório ao juntar ideias em simultaneidade temporal.

Futuro desenvolvimento: Implementar um modo de criação livre pode nos ajudar com insights sobre o que fazer para prosseguir. Elaboramos a hipótese de que ter a carta de edição sempre à mão seria útil para estes casos onde pequenos ajustes são necessários para que as ideias soem bem simultaneamente. Permitir ao jogador solicitar as cartas por tipo e descartar suas cartas indesejadas pode potencializar a liberdade criativa que se começou a sentir nesta etapa do projeto.

5.4 Investigação sobre IAs e composição através do Jogo *Carmim*

Nesta seção relatamos sobre a experiência de utilizar as ferramentas de IA da biblioteca *magenta.js* no jogo. Nos debruçamos sobre a experiência composicional como um todo e também analisamos cada uma das habilidades fornecidas (e implementadas) da biblioteca. Adicionalmente também refletimos sobre habilidades não fornecidas pela biblioteca e expectativas que surgiram durante o processo, mas que não foram atendidas.

5.4.1 As funcionalidades de IA implementadas no jogo

Analisamos separadamente cada funcionalidade fornecida pela biblioteca *magenta.js* e que foi implementada no jogo através de suas respectivas cartas a fim de relatar a experiência ao usar as ferramentas disponíveis para fins composicionais.

As ferramentas de continuação (*Carta de Continuação*) se provaram capazes de serem utilizadas criativamente com algumas ressalvas. A funcionalidade de continuação tem o potencial de gerar ideias cativantes, porém muitas vezes gerava ideias demasiadamente desconexas ou desinteressantes. Para contornar este problema foi necessário permitir ao jogador solicitar várias continuações até encontrar uma adequada.

As variações (*Carta de Variação*) geradas pela biblioteca mostram resultados consistentes quando o grau de variação é pequeno ou médio. Aumentar o índice de variação

pode gerar resultados um pouco menos consistentes em termos de qualidade, porém, via de regra, aceitáveis.

As habilidades de interpolação (*Carta de Interpolação*) entre duas sequências se mostraram adequadas. Entretanto, em alguns casos reparamos que há uma mudança muito acentuada, como uma espécie de cesura, em volta do índice de interpolação de 50%. Como exemplo, imaginemos a interpolação de uma ideia *A* com uma ideia *B*: durante a primeira metade da interpolação, as ideias são bastante parecidas com a ideia *A* e, na transição para a segunda metade, as ideias bruscamente se parecem com a ideia *B*. Uma melhoria possível para o jogo seria gerar mais etapas de interpolação para buscar um gradiente mais completo, com menos lacunas, entre as ideias. No futuro também podemos criar uma maneira de interpolar entre 4 ideias - funcionalidade fornecida diretamente pela biblioteca. Também se pode permitir misturar um número arbitrário de ideias, o que necessitará de uma implementação criativa em que se faz mais de uma etapa de interpolação¹⁸.

A geração de ideias musicais sem necessidade de nenhum *input* (*Cartas Geradoras*) é provavelmente o ponto fraco do conjunto de ferramentas que a biblioteca fornece. Utilizar esta funcionalidade implica muitas vezes em ter que requisitar diversas sequências até que uma delas seja útil/desejável e, além disso, não parece haver variedade suficiente (de métrica, perfil, harmonia, etc) dentre as ideias que geralmente acabavam sendo escolhidas para uso. As melodias geradas muitas vezes têm tessituras muito amplas e textura muito pontilhistas. É uma funcionalidade utilizável na forma atual, porém ainda de certa forma, limitada.

Ao relacionar estas observações com as habilidades que definimos na Seção 4, podemos dizer que a biblioteca teve mais sucesso com as habilidades de *Derivação* do que com as habilidades de *Criação direta*.

As cartas não seguem nenhum estilo específico. Esta funcionalidade ainda não foi implementada no jogo, mas existe na biblioteca. Os que assim desejam podem utilizar o modelo *MidiME* que foi feito para ser facilmente personalizável. Uma ideia para o futuro do jogo *Carmim* é implementar esta habilidade. Pode-se, por exemplo, permitir ao jogador entrar com uma ou mais ideias no início da partida que servirão como guia de estilo para as operações da *magenta.js*.

5.4.2 Funcionalidades **não** fornecidas pela biblioteca

Uma expectativa que tínhamos e que foi frustrada é sobre um ponto problemático e comum a todas as funcionalidades atuais da biblioteca: as ideias geradas todas têm a mesma métrica. Não se pode, por exemplo, gerar ideias em compasso ternário composto (ou com qualquer métrica com subdivisão ternária).

¹⁸ Um exemplo pode ser encontrado em *NSYNTH:...* (2021), em que por meia da concatenação de quadrados (que representam uma interpolação entre 4 timbres) uma rede (*grid*) de interpolações entre timbres foi criada a partir da concatenação destes quadrados. O resultado permite interpolar entre 64 timbres diferentes.

Durante as partidas notou-se que combinar as ideias verticalmente era sempre um desafio, uma vez que não havia nada que garantia que a simultaneidade das ideias geradas soaria de maneira harmoniosa (quando assim se desejava). Ou era necessário gerar novas ideias ao acaso até encontrar uma que soasse bem em simultaneidade com as outras ou então usar a carta de edição para fazer os ajustes finos a fim de satisfazer algum tipo de acorde, relação contrapontística, etc. Seria preciso algum tipo de habilidade de arranjo que pudesse juntar diversas ideias escolhidas para que soassem bem simultaneamente de acordo com os resultados esperados pelo compositor. Isto seria feito com a parametrização do método a ser aplicado nas sequências (blocos harmônicos, contraponto, conjunto de classes de altura, etc.). Igualmente poder-se-ia parametrizar decisões a serem tomadas acerca do ritmo e duração das sequências musicais envolvidas¹⁹.

¹⁹ A biblioteca fornece alguns **checkpoints** que permitem condicionamento por acordes, mas na nossa experiência os resultados não foram satisfatórios. Muitas vezes a solicitação de condicionar uma melodia a certo acorde parecia não ter efeito.

6 Magenta.js

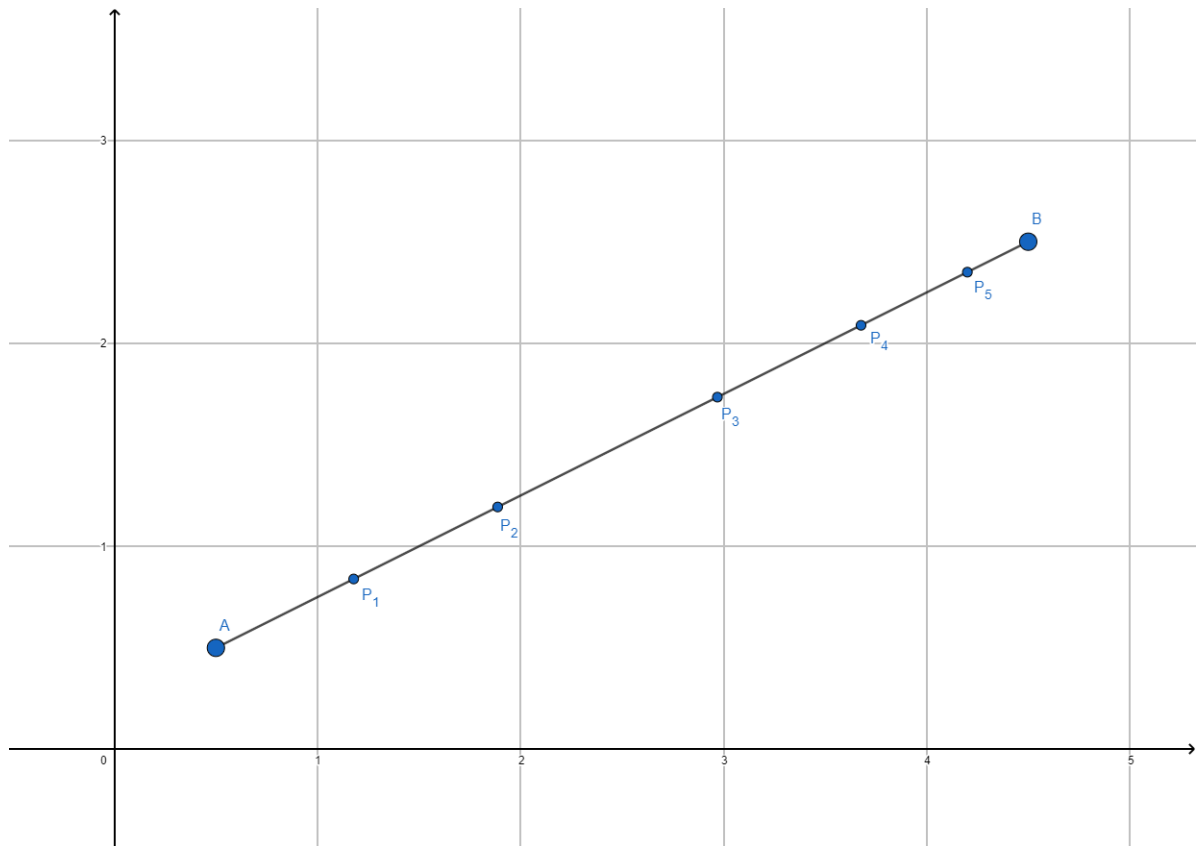
Esta seção descreve em um nível maior de detalhes as habilidades da biblioteca *magenta.js* que foi utilizada para o design do jogo *Carmim*. Assim como na Seção [Breve introdução às redes neurais](#), iremos apresentar de maneira superficial alguns conceitos subjacentes aos modelos utilizados com finalidade didática conforme descrito na Seção [Metodologia](#) e também com o objetivo de dar ao leitor uma intuição sobre as tecnologias que são tema de todo o texto. Antes de prosseguirmos, veremos os seguintes conceitos importantes: *interpolação linear* e *Variational Auto Encoders*.

6.0.1 Breve introdução matemática à Interpolação Linear

Introduziremos brevemente a técnica de interpolação linear para que posteriormente possamos descrever como ela é usada pela *magenta.js* para misturar melodias. Novamente muitos detalhes técnicos de como os modelos funcionam estão fora do escopo do presente trabalho e apresentamos apenas o que julgamos necessário para que um leitor leigo tenha uma intuição sobre o assunto, porém sem deixar alguns conceitos fundamentais de lado.

Utilizaremos como exemplo um plano cartesiano. Na Figura [11](#) apenas os pontos A e B são dados. O método de interpolação linear permite encontrar todos os pontos entre A e B . Na imagem destacamos, como exemplo, cinco pontos, P_1 até P_5 , dentre os infinitos pontos possíveis no segmento de reta $\overline{AB}$.

Figura 11 – Exemplo de interpolação linear



Dados apenas os pontos A e B os pontos de P_1 até P_5 foram obtidos por interpolação linear.

Os pontos pertencentes ao segmento de reta $\overline{AB}$ podem ser obtidos a partir da seguinte equação:

$$P = A + i \cdot \overrightarrow{AB} \mid 0 \leq i \leq 1 \quad (1)$$

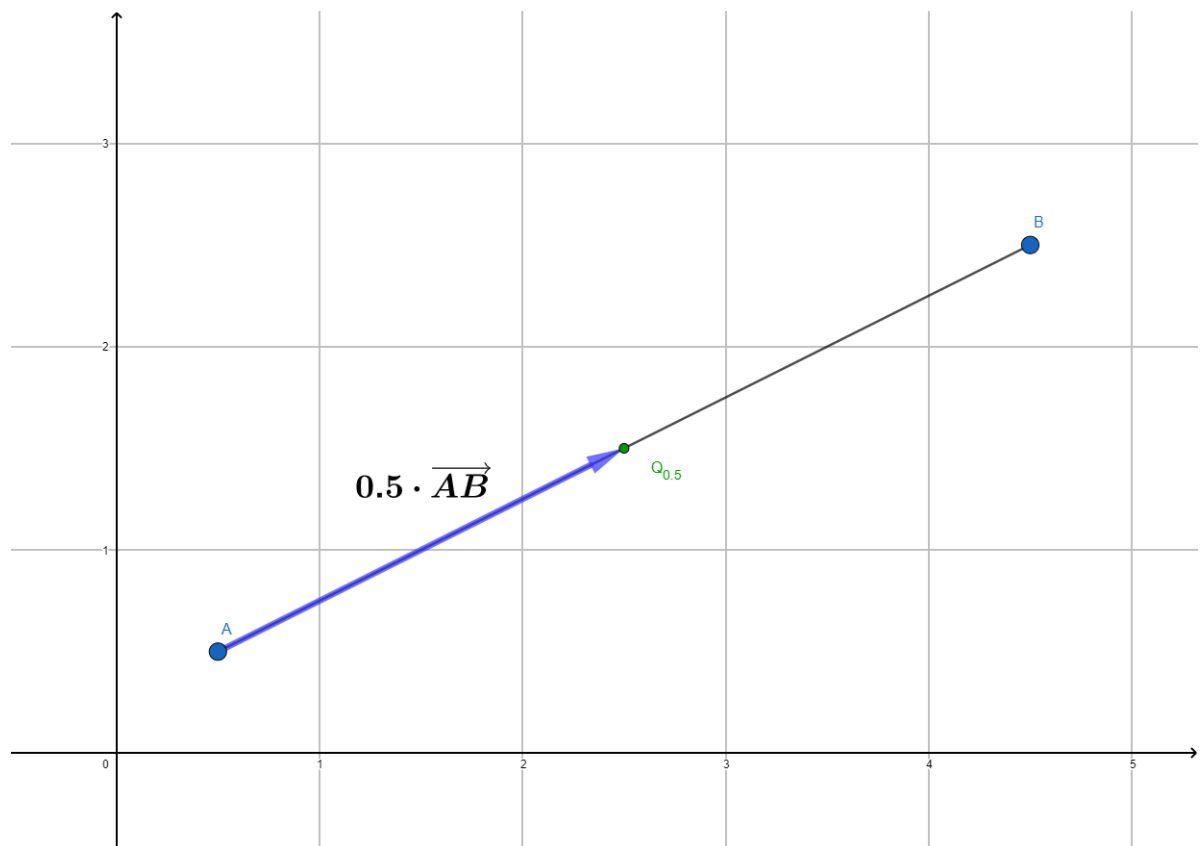
Onde

- P é o ponto a ser encontrado na reta $\overline{AB}$
- A é o ponto inicial da interpolação
- B é o ponto final da interpolação
- $\overrightarrow{AB}$ é um vetor capaz de “levar” A até B , tal qual $A + \overrightarrow{AB} = B$
- i é o fator de interpolação.

Aos leitores que não são familiarizados com *vetores* oferecemos uma breve explicação: imagine que o vetor é uma seta, com comprimento, direção e sentido. Esta seta não

tem nenhuma localização no espaço. Entretanto, ela pode ser aplicadas a um ponto para “transportá-lo”. É como se colocassem a base da seta no ponto de origem e ele fosse transportado para as coordenadas da ponta da seta. Um vetor pode ser escrito como $\overrightarrow{XY}$ que pode ser interpretado como “Um vetor que leva o ponto X exatamente ao ponto Y ”. Multiplicar um vetor por um número não altera sua direção, apenas sua magnitude e sentido (quando o fator da multiplicação é negativo). Quando um vetor é multiplicado por 2 seu comprimento dobra sem que sua direção e sentido se modifiquem. Multiplicar um vetor por -1 é efetivamente inverter o sentido do vetor enquanto se mantém a direção e comprimento. Multiplicar um vetor por 0.5 significa reduzir seu comprimento pela metade, e assim por diante. Este processo pode ser observado na Figura 12.

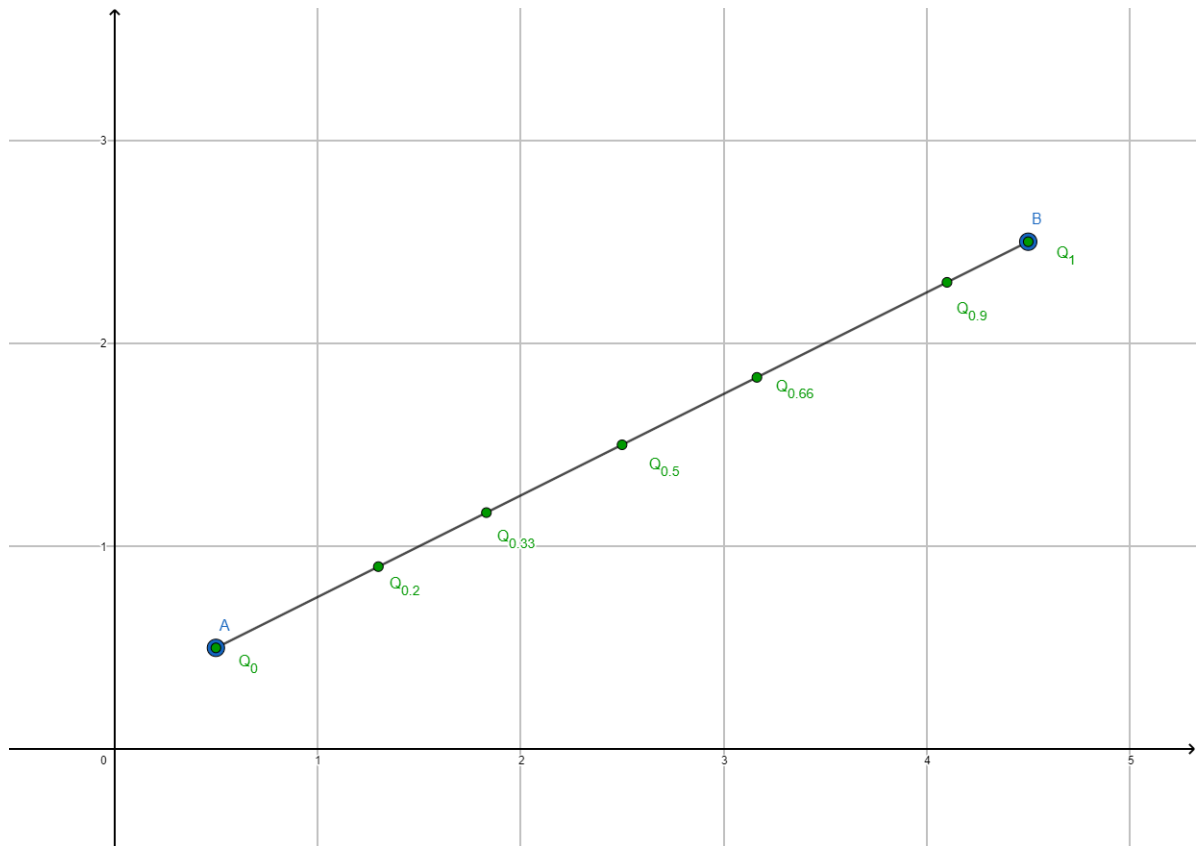
Figura 12 – Exemplo: encontrando o ponto médio de um segmento com interpolação linear



Quando $i = 0.5$ então P estará no ponto médio de $\overline{AB}$

Observe que quando $i = 0$ temos $P = A$ e para $i = 1$ temos $P = B$. Igualmente quando $i = 0.5$ então P estará no ponto médio de $\overline{AB}$ (Figura 12), quando $i = 0.33$ então P “percorrerá” $\frac{1}{3}$ do caminho entre A e B, quando $i = 0.9$ então P “percorrerá” 90% do caminho entre A e B, etc. (Figura 13).

Figura 13 – Exemplo de diferentes fatores de interpolação.

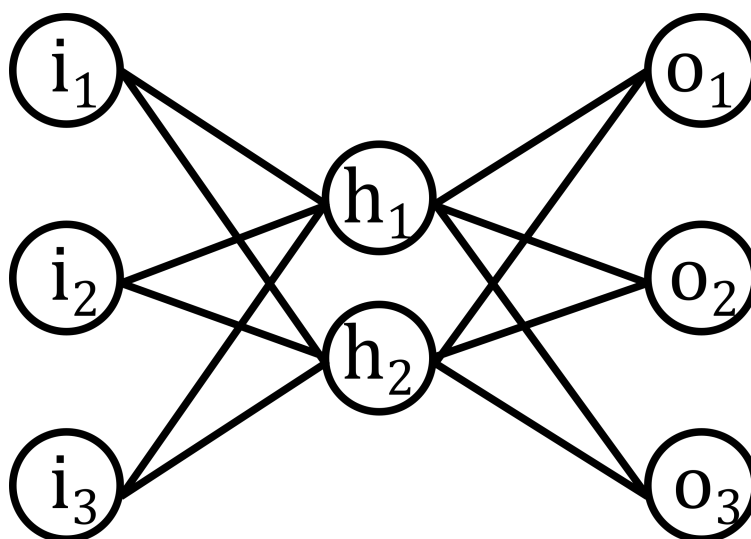


Interpolando 0%, 20%, 33%, 50%, 66%, 90% e 100% do caminho entre A e B

6.0.2 *Variational Auto Encoders*

Para entender como funciona um *Variational Auto Encoder* (VAE) primeiramente nos debruçamos sobre o conceito de *Auto Encoder* (AE). Nos baseamos em [Rocca e Rocca \(2019\)](#) ao elaborar esta seção do texto.

Figura 14 – Modelo básico de um Auto Encoder



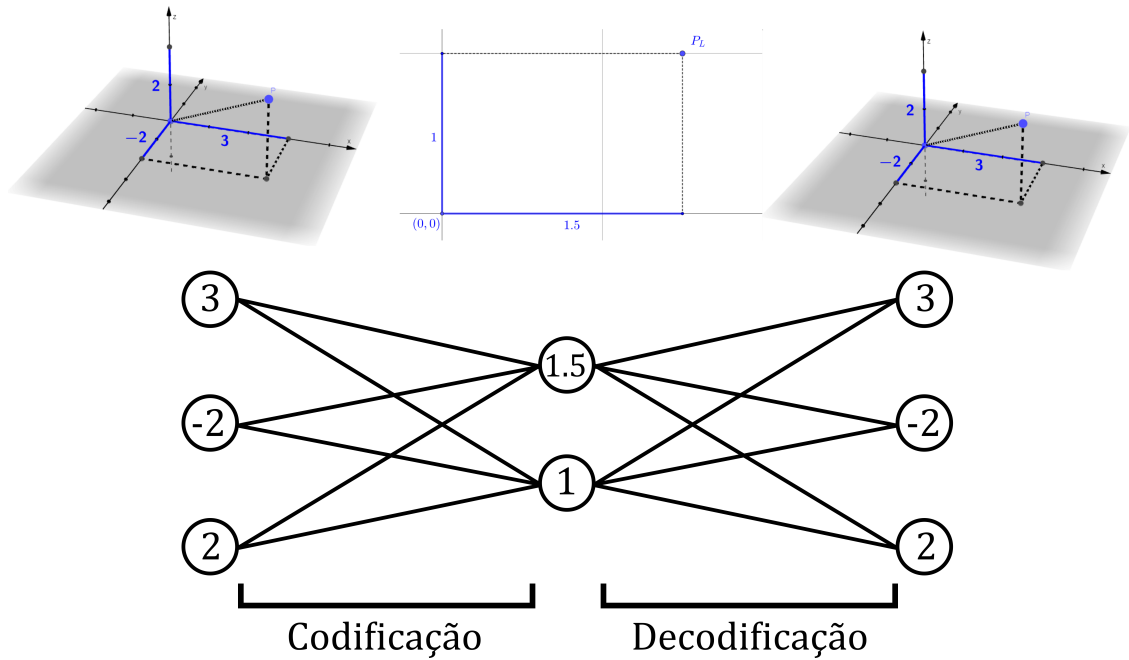
Na imagem temos uma rede com 3 neurônios na camada de entrada (i_x), 2 neurônios em uma única camada oculta (h_x) e 3 neurônios na camada de saída (o_x).

No exemplo (Figura 14) temos um esquema básico de um AE. Na camada de entrada (i) temos três números. Na camada oculta (h) temos 2 e na camada de saída (o) temos 3, igualando a camada de entrada. O objetivo de um AE é a *redução de dimensionalidade*: a ativação da camada oculta (h) em função da camada de entrada (i) contém um número menor de dados, porém pode ser armazenada e utilizada posteriormente para reconstruir os dados originais de entrada na camada de saída (o). Esta é uma forma de compressão e descompressão de dados. Dizemos que estas representações comprimidas de dados pertencem ao *espaço latente* do modelo.

A etapa de transformar os dados de entrada em uma representação no espaço latente é chamada de *codificação*. A operação inversa de recuperar a informação original do espaço latente, com ou sem algum grau de erro, é chamada de *decodificação*. Estes conceitos são ilustrados na Figura 15.

Quando a rede é capaz de fazer a decodificação exatamente igual aos dados codificados, dizemos que houve *compressão sem perda de dados*, e, caso contrário, *compressão com perda de dados*. O treinamento de um AE consiste justamente em minimizar o erro do par codificador-decodificador.

Figura 15 – Exemplo de redução de dimensionalidade em um *Auto Encoder*



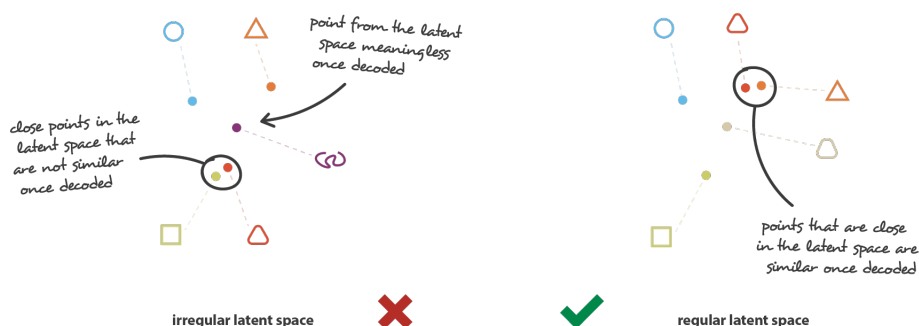
Podemos interpretar as camadas de entrada e saída como pontos em 3 dimensões, e pontos no espaço latente como em 2 dimensões. O mesmo raciocínio é válido para camadas com mais de 3 neurônios, porém sendo seres tridimensionais, apenas podemos visualizar de maneira intuitiva pontos em 3 ou menos dimensões. Entretanto, apesar de não conseguirmos visualizá-los, espaços com mais de 3 dimensões existem matematicamente e suas leis geométricas são equivalentes ao 2D e 3D, porém com mais dimensões. ([EUCLIDEAN...](#), 2021)

Um *Variational Auto Encoder* (VAE) é um tipo específico de AE que é geralmente utilizado para processos generativos (como o processo da *magenta* de gerar melodias). Por design, durante seu treinamento o espaço latente de um VAE é regularizado e uniformizado a fim de apresentar duas propriedades:

1. **Continuidade:** pontos próximos no espaço devem ser decodificados em resultados semelhantes. Isto implica, por exemplo, que dois pontos em proximidade no espaço latente serão decodificados em duas melodias similares. Ilustrada na Figura 16.
2. **Completeness:** pontos amostrados no espaço devem produzir resultados significativos. Isto implica que não há “buracos” no espaço latente cuja decodificação traria resultados que não representam adequadamente o tipo de dado modelado pela rede. Ilustrada na Figura 17.

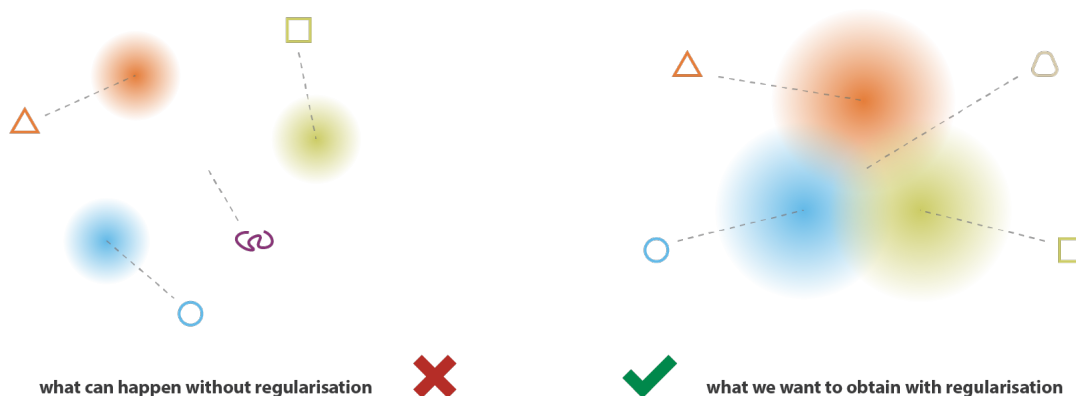
Tentaremos fazer com que as utilidades destas propriedades fiquem mais evidentes nas seções a seguir, onde detalharemos como a *magenta* realiza algumas tarefas, relacionando as propriedades com as ditas tarefas.

Figura 16 – Continuidade no espaço latente



Fonte: Rocca e Rocca (2019)

Figura 17 – Completude e não-completude no espaço latente



Fonte: Rocca e Rocca (2019)

A seguir, daremos uma visão de como o modelo *Music VAE* da Magenta realiza algumas de suas operações.

6.1 Amostragem

Para gerar melodias ou ritmos ([Criação direta](#)) basta escolher aleatoriamente um ponto qualquer pertencente ao espaço latente. A propriedade de **completude** garante que qualquer ponto escolhido será decodificado em um resultado significativo, representativo do conjunto de dados utilizado para treinar a rede neural.

6.2 Interpolação

A mesclagem entre 2 ou 4 sequências de notas musicais é feita com a técnica de interpolação linear (2 ideias) ou bilinear (4 ideias). Veremos apenas o caso da interpolação linear, pois é a única implementada no nosso jogo e porque o raciocínio da interpolação bilinear é similar, porém com duas dimensões (INTERPOLAÇÃO... , 2021).

A propriedade de **continuidade** faz com que pontos próximos no espaço latente representem ideias semelhantes. Cria-se assim regiões no espaço latente onde se pode mover em passos pequenos a fim de obter ideias similares. A propriedade de **completude** faz com que estas regiões todas se toquem, fazendo com que o espaço latente seja um verdadeiro gradiente no qual se pode passear e todos os pontos amostrados representarão melodias que evoquem o *training set*.

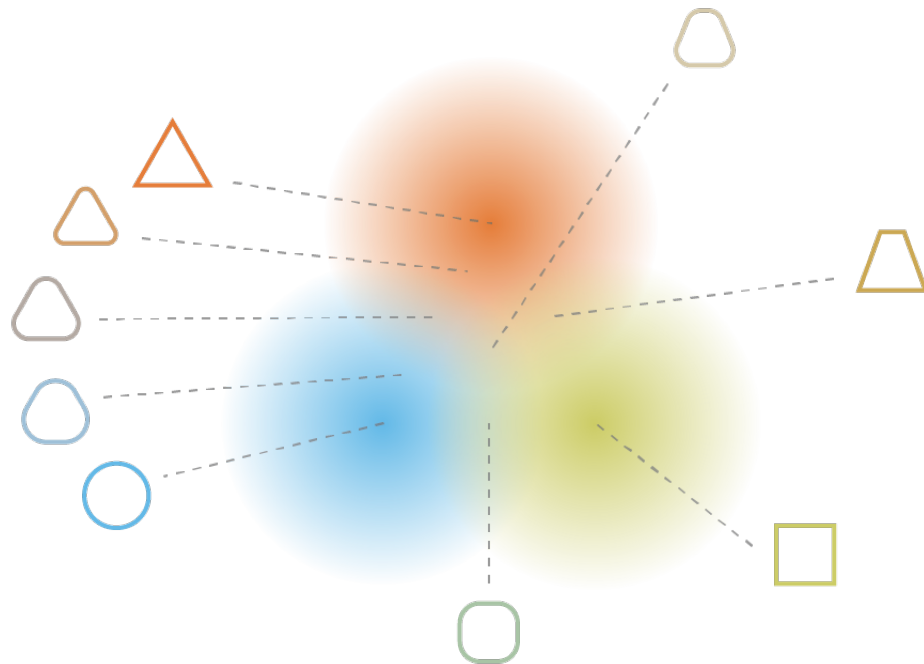
Para visualizar como o processo ocorre na rede tomemos um exemplo: interpolar entre duas melodias. Tal processo se daria:

1. Ambas as melodias são codificadas em pontos no espaço latente (A e B, por ex.);
2. De acordo com o índice de interpolação, encontra-se o ponto interpolado P pertencente à reta $\overline{AB}$;
3. O ponto P é decodificado e se obtém a melodia interpolada.

Tal processo pode ser executado iterativamente com índices de interpolação 0, 0.1, 0.2, ... , 0.9, 1 para gerar 9 melodias que são misturas das originais. Este processo é efetivamente caminhar em etapas pelo espaço latente entre os pontos que representam a primeira e a segunda melodias.

A combinação das propriedades de **continuidade** e **completude** fazem com que, durante este caminho, cada pequeno passo resulte em uma pequena mudança no resultado, e que cada passo cairá em uma ideia representativa do *training set*. Isto pode ser observado na Figura 18.

Figura 18 – Completude e continuidade no espaço latente



Fonte: Rocca e Rocca (2019)

6.3 Variação

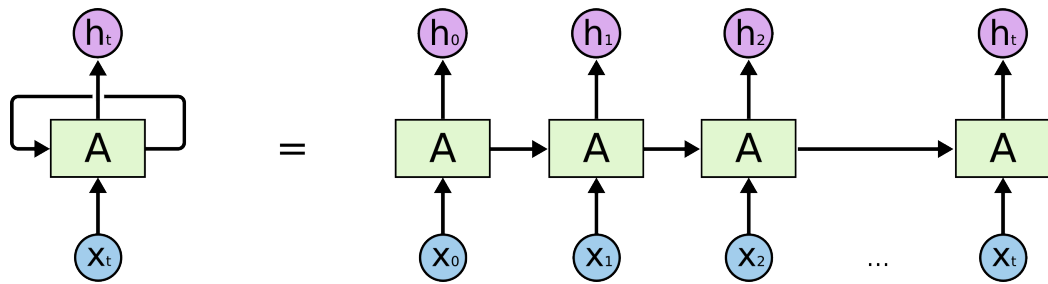
A propriedade de **completude** diz que pontos próximos no espaço são decodificados em resultados semelhantes. Isto significa que se pode variar uma sequência de notas da seguinte maneira:

1. A sequência de notas é codificada e se obtém sua representação no espaço latente;
2. Desloca-se uma distância desejada no espaço latente em uma direção aleatória. Quanto maior a distância, menos semelhante será o resultado da etapa seguinte e vice-versa;
3. Realiza-se a decodificação do novo ponto do espaço latente obtido na etapa anterior.

6.4 Redes recursivas e a habilidade de continuação

A Magenta fornece sua habilidade de continuação de sequências de notas através de um outro tipo de rede neural: a rede neural recursiva, ou **RNN**, do inglês *Recursive Neural Network*. Este tipo de rede é mais adequada para lidar com a tarefa de prever os próximos elementos de dados sequenciais. Nos baseamos em Olah (2021) para escrever esta seção.

Figura 19 – Exemplo de RNN



Fonte: Olah (2021)

Na Figura 19 podemos observar um esquema simplificado de uma RNN. Voltemos nossos olhos ao lado esquerdo da imagem, antes do sinal de igual. Os quadrados A representam as camadas da rede neural. Os círculos com a letra x representam os dados sequenciais de *input* e os círculos com a letra h , os dados sequenciais de *output*. De A saem duas setas: a primeira leva para um círculo com a letra h e representa o *output* normal da rede, ativação da camada de saída; a segunda seta sai de A e volta para A e representa um fluxo de saída de dados do *estado oculto* da rede. Este estado é um conjunto de números armazenados internamente na rede. A rede *RNN* é ativada iterativamente, em sequência, processo que é, com efeito, como se a rede tivesse sido copiada e ativada em sequência, como ilustrado na parte direita da imagem. A cada etapa, o *estado oculto* da rede influencia na execução atual e é modificado em função dela.

A Magenta usa um tipo específico de rede neural chamada LSTM (do inglês “*Long Short Term Memory Networks*”). As peculiaridades técnicas das LSTMs são bastante complexas então nos limitaremos a citar sua especialidade: aprender relações de longo prazo entre os dados, quer dizer, relações entre dados muito distantes nas sequências. Por exemplo: a relação entre notas que apareceram no início de uma melodia e as notas que aparecerão vários compassos depois.

Quando a Magenta precisa gerar a continuação de uma melodia ou ritmo, os dados de entrada são passados sequencialmente para a rede. A cada etapa, o *estado oculto* da rede se modifica. Ao chegar no último elemento dos dados de *input*, a rede neural utilizará este elemento como dado de entrada da próxima iteração e, com seu *estado oculto* afetado pela sequência de entrada, continuará o mesmo processo. Entretanto, desta vez cada nota recém gerada da continuação será utilizada como entrada da rede para descobrir a próxima. O processo é interrompido arbitrariamente de acordo com o tamanho da continuação requisitada.

7 Uso das ferramentas de IA fora da academia

Nesta seção discutiremos os usos criativos das ferramentas de IAs por artistas que, de algum modo, estão fora do meio acadêmico. Buscamos com isto entender a relação destas tecnologias relativamente recentes com o corpo geral de artistas e com isso nos aproximar de um melhor entendimento sobre as diversas questões que permeiam a interação entre IA e música.

Isto não significa que não houvesse nenhum pesquisador e/ou técnico da área envolvido com os artistas (apesar de que, em alguns casos, não havia), mas sim que estes artistas não teriam algum tipo de “motivação acadêmica” em utilizá-las: seja por questões institucionais acadêmicas (encorajamento em constantemente produzir conhecimento) ou até por ver no próprio problema uma arte:

[...] existe motivação, força motriz humana que deseja que isso [*desenvolvimento de uma IA capaz de compor genuinamente*] aconteça. Forças estas que não estão orientadas por uma visão mercadológica, objetivando retorno financeiro, mas sim advindas dos mesmos agentes que têm se dedicado à composição algorítmica estrita, e que veem na própria formalização algorítmica um campo de expressão, uma arte. [...] (BUCK, 2018, p. 90–91)

Buscamos trabalhos que envolvem artistas que possam olhar para as ferramentas de IA e, sem tais motivações, se concentrar em questões práticas de criação: o que estas ferramentas têm para me oferecer como artista? O que eu posso/quero fazer com isto?

7.1 YATCH

Em uma reportagem do site Ars Technica, Mattise (2019) nos conta sobre como a banda YACHT escreveu seu álbum *Chain Tripping* utilizando as ferramentas de IA da *Magenta*, especialmente seu modelo *MusicVAE*. Nenhum dos integrantes da banda, apesar de familiarizados com tecnologia, é programador (ou da área de ciência de dados):

“We may be a very techy band, but none of us are coders,” Evans tells Ars. “We tend to approach stuff from the outside looking in and try to figure out how to manipulate and bend tools to our strange specific purposes. [...]”

A banda utilizou exclusivamente as ferramentas da biblioteca para gerar as sequências de notas das músicas. As letras também foram feitas com IA através de uma colaboração com Ross Goodwin²⁰, escritor que se descreve como um “poeta de dados” (*data poet*). Os integrantes contam ter se limitado propositalmente a usar apenas ideias geradas pelas IAs, sem, por exemplo, improvisar sobre os dados de saída dos algoritmos.

Na entrevista a banda segue falando sobre suas intenções:

²⁰ Os modelos/ferramentas utilizadas não constam na reportagem.

“AI seemed like an almost impossible thing, it was so much more advanced than anything we had dealt with... And we wanted to use this to not just technically achieve the goal of making music—so we can say, ‘Hey an AI wrote this pop song’—rather we wanted to use this tech to make YACHT music, to make music we identify with and we feel comes from us.”

Muito pertinente aos nossos questionamentos é o desejo dos integrantes de continuar realizando um processo de criação bastante humano e artesanal:

“It still couldn’t make a song just by pushing a button; it was not at all an easy or fun flow to work through,” Bechtolt says. [...] “We’d have to be involved. There’d have to be a human involved at every step of the process to ultimately make music... The larger structure, lyrics, the relationship between lyrics and structure—all of these other things are beyond the technology’s capacity, which is good.”

É uma visão da *IA* como ferramenta a serviço do compositor. Ferramenta capaz de ampliar os horizontes criativos, trazendo possibilidades que, sem ela, talvez fossem inalcançáveis. Sobre um interlúdio instrumental da música *SCATTERHEAD*, o integrante *Jona Bechtolt* disse:

“But I don’t know if we could’ve written it ourselves. It took a risk maybe we aren’t willing to take when we’re writing a pop song, and it ended up in a place that’s really interesting and beautiful.”

Para atingir o objetivo de criar música com a qual eles se identificavam, a banda treinou o modelo *IA* com materiais de um catálogo de ideias musicais que eles já tinham construído²¹. Curiosamente a integrante *Claire L. Evans* comenta sobre as capacidades generativas da ferramentas:

“These tools, because they’re failing at doing exactly what they’re supposed to do, they tend to fall just outside the boundary of what’s an acceptable, normal, or traditional melody,” Evans says. “It’s just on the other side of the fence, close enough that we can see it and say, ‘That’s a sentence, that’s a melody.’ It’s not completely formless, it’s just outside of what humans would do. And that space between meaning and meaninglessness and what’s expected and what’s out of bounds is such an interesting space.”

Vê-se que se reconhece uma limitação nas ferramentas (pelo menos no quesito de imitar o comportamento humano) mas, ao mesmo tempo, tal característica é vista como interessante *per se* e uma qualidade potencialmente desejável, como corroborado pelo comentário seguinte de Evans:

“This tech is evolving so quickly, by the time this record is maybe a year old, a lot of the stuff we appreciated for its wonkiness and strangeness will be so polished. They’ll produce melodies and lyrics that are unrecognizable as machine generated, they’ll be ‘perfect’.” Evans says.

²¹ Interpretação de “Back catalogue”, como disseram os integrantes.

“Something will be really lost, because when it’s not perfect there’s this specialness unique to our moment in time. We’re going to be nostalgic. People will come back to this slightly undefinable wonky aesthetic that AI has—not just in music, in images, text—and it’ll become the new analogue. In the way musicians fetish lo-fi, writers use typewriters, or we buy vinyl instead of just streaming, all these nostalgic feelings will be associated with this moment in time for AI.”

A popularização de ferramentas amigáveis de *IA* direcionadas para um público músico no geral é um processo relativamente recente e ainda em progresso:

YACHT had long possessed an interest in AI and its potential application in music. But the band tells Ars it wasn’t until recently, around 2016, that the concept of doing a full album using this approach seemed feasible. While research entities had long been experimenting with AI or machine learning and allowing computers to autonomously generate music, the results felt more science project than albums suitable for DFA Records [...]. Ultimately, a slow trickle of simplified apps leveraging AI [...] finally gave the band the idea that now could be the time.

7.2 Shadow Planet

Vincent (2021) escreveu em sua reportagem ao website *The Verge* sobre a colaboração entre o escritor *Robin Sloan* e o músico *Jesse Solomon Clark* que co-criaram com a ferramenta *Jukebox*²² o álbum *Shadow Planet*. Apesar de Robin não ser da área de ciências de dados e se descrever como escritor, diz ter habilidades de programação.

Sloan também pensa em *Jukebox* apenas como uma ferramenta, porém vai além e a descreve comparando-a diretamente a um instrumento musical:

Making this album, I learned that this kind of AI model is absolutely an “instrument” you need to learn to play. And I’ve come to believe that analogy is a lot more useful and generative than like, “AI co-composer” or “automatic AI artist” or whatever other analogy you might have heard or can imagine. It’s basically a tuba! A very... strange... and powerful... tuba...

Na entrevista Robin comenta que apesar da dificuldade intrínseca do processo criativo, um fator motivacional muito forte foi a força evocativa e o interesse dos materiais resultantes. Algumas características limitantes foram abraçadas e incorporadas na obra como a característica *Mono* do material gerado além de uma qualidade *etérea* do timbre, que soa como se alguém estivesse sintonizando uma rádio distante.

Uma nota paralela sobre outro trabalho de Sloan: o autor ao escrever o audiolivro de sua obra *Sourdough* utiliza uma implementação da *IA SampleRNN* para criar, a partir de variações e distorções de obras corais croatas (*klapas*), trechos musicais de sonoridade excêntrica, evocando uma língua inexistente criada para o livro que era cantada na cultura fictícia dos *Mazg*. O autor comenta como esta limitação em reproduzir corretamente as

²² Criada pela *Open AI*.

linguagens existentes foi decisiva para a criação de um áudio na língua fictícia dos *Mazg* (SLOAN, 2021). O autor precisava de algo que soasse familiar, como algo realmente crível, porém que não pudesse realmente ser identificado como próximo de uma linguagem existente - exatamente naquele espaço estranho entre o artificial e o humano descrito pelos membros da banda *YACHT*.

7.3 Hello World e FlowMachines

O artista *SKYGGE*²³ descreve seu álbum *Hello World* como o primeiro álbum escrito por um artista utilizando *IA*²⁴ (HELLO..., 2021). O álbum foi consequência da incorporação de artistas ao projeto de pesquisa *The Flow Machines project*²⁵. Além do uso das ferramentas do *FlowMachines*, o álbum foi criado com a colaboração de diversos outros artistas.

A ferramenta utilizada para a criação de *Hello World* foi o *FlowComposer*, a primeira ferramenta composicional assistida por *IA* que gera *lead sheets* (PACHET; ROY; CARRÉ, 2021). Diversos outros artistas já criaram álbuns inteiros ou composições utilizando *FlowComposer* (PACHET; ROY; CARRÉ, 2021, pg. 8–22). Esta *IA* demonstra grande versatilidade uma vez que pode ser treinada em qualquer material de interesse dos compositores, permitindo que cada um consiga personalizar as ferramentas de acordo com suas preferências e interesses. O *FlowComposer* também já foi utilizado para diversas outras atividades além da composição do álbum *HelloWorld*, incluindo concertos cujas composições foram realizadas ao vivo. (PACHET; ROY; CARRÉ, 2021, p. 14–15).

O álbum ganhou vários remixes tradicionais (no sentido de não utilizarem *IA*). Fato que pode ser interpretado como um sinal da qualidade do trabalho com *IA* (PACHET; ROY; CARRÉ, 2021, p. 22).

As ferramentas desenvolvidas para o projeto também têm a capacidade de realizar harmonizações de melodias de *input* no estilo desejado. Uma harmonização da canção *Começar De Novo* de *Ivan Lins* foi feita ao estilo do grupo vocal *Take 6* e apresentado ao próprio compositor que reagiu de maneira muito entusiástica (PACHET; ROY; CARRÉ, 2021, p. 5).

O projeto *FlowMachines* teve vários desdobramentos não finalizados, dos quais destacamos: a criação de um software de composição para crianças; geração de música especificamente para trilhas sonoras; um software (*Reflexive Looper*) que atua como acompanhador, capaz de improvisar/reagir em tempo real ao performar com um músico humano²⁶;

²³ Nome artístico do compositor Benoit Carré para englobar seu trabalho com *Flow-Machines*.

²⁴ “*Hello World*” é o exemplo canônico de texto utilizado em tutoriais de programação como a primeira etapa no aprendizado de qualquer linguagem.

²⁵ Conduzido pela Sony Computer Science Laboratories e universidade de Sorbonne (2012-2017) com subsídio do Conselho Europeu de Pesquisa (ERC).

²⁶ Nesta área um competidor muito forte é o software *Omax* do *IRCAM* e suas ramificações.

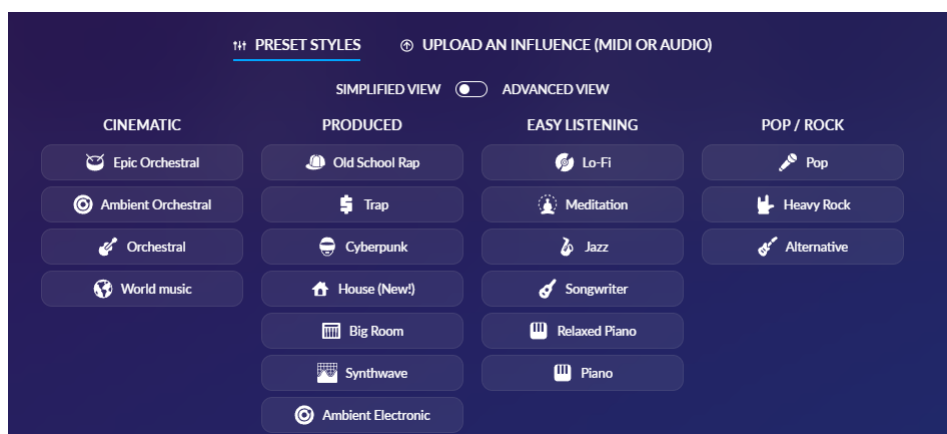
Max Order: A Tale of Creativity, uma *Web Comic* escrita por uma *IA*, cuja personagem principal é uma artista que representa a própria *IA* que a escreveu.

O projeto *Flow Machines* atualmente segue conduzido pela *Sony CSL* e continua contando com a participação de artistas com carreiras (e famas) bem estabelecidas, como Jean-Michel Jarre.

7.4 O problema do licenciamento

Há ainda o uso de ferramentas por pessoas da área de criação ou áreas relacionadas com o fim de atender demandas comerciais/profissionais. Por exemplo: a *IA* AIVA é capaz de criar composições numa variedade impressionante de estilos (Figura 20) com poucos cliques. Além disso, o usuário pode fornecer como *input* suas próprias influências musicais tanto em formato de áudio quanto MIDI. Um dos atrativos comerciais da AIVA é a facilidade que traz à criadores em obter músicas licenciadas para diferentes usos. No seu *website* podemos ver uma propaganda que foca diretamente neste ponto. Atualmente há 2 planos pagos: o plano *Standard*, no qual o usuário *não* detém os direitos autorais das obras criadas pela *IA* e pode usar suas músicas (sem necessidade de dar créditos) exclusivamente nos websites *YouTube*, *Twitch*, *TikTok* e *Instagram*; o plano *Pro* no qual o usuário tem direitos autorais sobre as composições geradas. Empresas no plano *Pro* precisam atender a um critério de limite de renda anual. Caso ultrapassem o dito limite, deverão fazer um contrato diretamente com a corporação responsável pela AIVA.

Figura 20 – Interface de seleção de estilos da AIVA



A imagem mostra a impressionante variedade de estilos em que AIVA é capaz de compor. (AIVA:..., 2021)

Tal ferramenta pode ser usada tanto por artistas não-músicos de diversas áreas que buscam trilhas sonoras para seus projetos (games, cinema, teatro, etc.) como por “cri-

adores de conteúdo”²⁷ e até mesmo músicos compositores que precisam lidar com uma demanda comercial muito alta de produção de obras (considerando que assine o plano adequado).

A existência de tais planos e licenças implicam na existência de uma base legal para que uma *IA* (ou seus criadores) possa possuir direitos autorais sobre suas obras. Há, entretanto, questões em aberto com relação a como humanos reagirão a isto: como reagiria um cliente ao descobrir que o compositor que contratou para criar uma trilha sonora entregou uma obra feita por uma *IA*? Tais ferramentas ainda não são abundantes e difundidas o suficiente para tal questionamento realmente vir à tona. Além disso, a reação de tal cliente pode ser diferente em distintos estágios de evolução da tecnologia, dependendo do quão próximo do potencial humano esta chegou. Adicionalmente, conjecturamos que se este compositor usou a ferramenta apenas para gerar o material inicial que posteriormente serviu como base para seu trabalho, a reação provavelmente seria mais positiva.

7.5 Considerações sobre a adoção e impacto da *IAs* por artistas fora do circuito acadêmico

Encontramos obras feitas por artistas que tiveram a iniciativa de buscar as ferramentas de *IA*, como o caso da banda YACHT. Vimos o caso do Projeto FlowMachines no qual pesquisadores e artistas trabalharam de maneira íntima durante todo o processo.

Notamos um fato peculiar: a banda *YATCH* usou os *Collab Notebooks* da Google para compor o álbum *Chain Tripping* (MATTISE, 2019), ferramenta esta normalmente destinada a programadores²⁸. Isto nos levou a perceber que além do *Carmim* poder ser uma ferramenta de investigação e pesquisa, ele também tem potencial para ser uma ferramenta composicional autônoma. Apesar de necessitar de melhorias, por possuir uma interface simplificada e (possivelmente) mais intuitiva, ele também tem potencial para permitir a artistas utilizar as ferramentas de *IA* com mais facilidade do que por outros meios disponíveis. Ainda relacionando a experiência da banda com nossa experiência ao utilizar o *Carmim*, algumas funcionalidades adicionais ainda parecem desejáveis no nosso jogo. Os integrantes de *YACHT* descrevem um impecilho que encontraram durante seu trabalho:

“So we had to notate all our songs in MIDI, which is a laborious process,” Bechtolt continues. “We have 82 songs in our back catalog, which is still not really enough to train a full model, but it was enough to work with the tools we had.” (MATTISE, 2019)

²⁷ Como são chamados os usuários de redes sociais que criam mídia que atrai usuários para a rede, geralmente com grande quantidade de seguidores e que obtém retorno financeiro por sua presença nas plataformas sociais.

²⁸ Apesar dos cadernos deixados pelo time da *magenta.js* deixar alguns elementos de interface gráfica como botões, etc. destinados ao público geral.

Implementar no jogo as ferramentas do modelo *Onset and Frames* da *magenta.js*, modelo capaz de transcrever áudios de piano em dados *MIDI* poderia agilizar este tipo de processo. Entretanto, pode-se argumentar que utilizar um controlador *MIDI* (como um piano digital, por exemplo) para fazer o *input* dos dados diretamente no formato final é mais eficiente.

Implementar o modelo *SPICE*, capaz de estimar as alturas das notas de um áudio, pode ser útil na ausência de um controlador *MIDI*, além de que pode ser útil para pessoas que não têm familiaridade com instrumentos de teclado ou teoria e percepção musical, pois permite ao usuário cantar as melodias e em seguida lhe apresentaria uma estimativa das alturas das notas.

Além das ferramentas feitas para realizar tarefas diretamente ligadas à criação, também há ferramentas que foram feitas para realizar tarefas mais manuais/repetitivas que por vezes fazem parte de um processo criativo, porém não estão ligadas diretamente à criação como é o caso do [Reconhecimento ótico de partituras](#)²⁹. *IAs* também conseguem realizar outras atividades relacionadas (como [Acompanhamento de partitura \(Score Following\)](#)) que têm aplicações tanto práticas (no sentido de automatizar uma tarefa) quanto criativas.

Observamos exemplos de artistas que encontraram nas ferramentas de *IA* novas possibilidades, não apenas pela capacidade de realizar as tarefas de criação à qual estas máquinas se propõe, mas também pelas suas imperfeições e limitações. Artistas encontraram neste espaço ambíguo entre o humano e o artificial um potencial criativo único, que talvez seria até inalcançável com ferramentas que soassem demasiadamente humanas (como no caso dos cantos dos *Mazg* de Robin Sloan).

Durante nossas investigações notamos que as *IAs* que funcionam como ferramentas auxiliares aos artistas aparentemente receberam uma atenção maior do que as ferramentas destinadas a compor autonomamente, sem a participação de um humano. Na definição apresentada por [Buck \(2018, p. 1–4\)](#), as ferramentas de *Composição Assistida por Computador (CAC)* receberam uma atenção maior do que as ferramentas de *Composição Gerada por Computador (CGC)*. Tal percepção³⁰ aparece numa crítica (ou *Observation*) do *Scientific American* ([PACHET; ROY; CARRÉ, 2021, p. 28](#)).

Suposta predileção por estas ferramentas que requerem intervenção humana pode estar relacionada ao efeito *IKEA*³¹, que faz com que tenhamos percepções desproporcionalmente positivas sobre coisas que criamos, mesmo que parcialmente ([PACHET; ROY; CARRÉ, 2021, p. 29](#)). Corroborando esta visão encontramos o seguinte relato do compositor *Alex Buck*:

²⁹ É sensato supor que ferramentas que automatizem tais processos repetitivos e entediantes são quase unanimemente bem aceitas e recebidas.

³⁰ Certamente ainda subjetiva - aqui caberia uma análise numérica detalhada, o que merece uma investigação própria.

³¹ Loja famosa por vender mobílias prontas para serem montadas pelo próprio consumidor.

Por outro lado, quando utilizei o Autocousmatic, programa que é capaz de gerar uma composição inteira, pronta em minutos, não senti a mesma relação com o resultado. Alimentar o programa com um banco de sons e deixá-lo realizar todo o trabalho, sobretudo em se tratando de um programa em que só tive o trabalho de baixá-lo da internet e que não me oferecia nenhuma possibilidade de escolha para além do número de canais e da duração da peça, não foi suficiente para me satisfazer nem esteticamente, nem para me sentir autor da mesma. Poderia utilizar um trecho interessante do resultado gerado pelo algoritmo, mas jamais a peça inteira. Esse sentimento de, efetivamente, não estar compondo quando o poder de escolha e determinação de processos está circunscrito apenas ao apertar de um botão é um indicativo de que o ser humano não investirá em substitutos para ofícios ligados às ciências, artes e filosofia. (BUCK, 2018, p. 90)³²

É importante notar que *Autocousmatic* não usa *aprendizado de máquina*, porém isto não prejudica o argumento. Conjecturamos que se fosse uma *IA* possivelmente a percepção das pessoas sobre usar o software para criar composições inteiras seria ainda mais negativa dada a (má) fama que estes softwares têm de substituir humanos nas mais diversas áreas. De certo ponto de vista, ao ignorar detalhes de implementação, uma *IA* continua sendo um software com capacidades definidas e o usuário pode ser indiferente à maneira com a qual o software atinge seu objetivo. Possivelmente, quando as capacidades das *IAs* se igualarem ou se aproximarem às capacidades humanas, este cenário pode mudar.

7.5.1 O potencial de apropriação de todo software

Ainda sobre ferramentas ditas “autônomas” que geram composições inteiras há de se considerar o potencial de parametrização e customização que existe em (literalmente) todo *software*³³:

Por exemplo, *Autocousmatic* (BUCK, 2018, p. 72–75) utiliza um banco de cinco composições - feitas por humanos - como base para a parametrização de seu processo de criação. O processo de escolha destas 5 obras tal qual o processo de criação do algoritmo que “as usa como inspiração” é, sob certa ótica, similar ao processo de escolha e preparo de um *training set* na hora de treinar uma inteligência artificial. O fato destas cinco composições serem fixas no *Autocousmatic* é apenas uma limitação de sua implementação e, uma vez circunscrita, os usuários poderiam customizar obras de *input*. Também deve-se observar que fica sob escolha e responsabilidade dos desenvolvedores criar (ou não) para o usuário final uma interface que permite a ele controlar os parâmetros/variáveis dos sistemas envolvidos.

Constata-se ainda que todo software pode ser publicado como *software de código aberto* e/ou *livre*, o que permite que outras pessoas (com a devida competência técnica) consigam

³² O software ainda permite um grau (mesmo que limitado) de envolvimento humano não contemplado na citação (porém contemplado no texto original): apesar de não ter controle sobre como serão usadas, o usuário fornece como entrada algumas amostras de som para o software.

³³ Potencial este que **pode ou não** ser aproveitado dependendo de questões comerciais, éticas, filosóficas, práticas, de capacidade técnica, etc.

criar bifurcações (*forks*)³⁴ do código e customizar o próprio algoritmo do software, ou seja, adaptar a ferramenta para seus interesses de dentro pra fora. Neste sentido, todo software tem um potencial - *nem sempre aproveitado* - de ser customizável para os processos criativos individuais dos artistas.

Assim sendo, apesar de suposta predileção dos artistas em torno das ferramentas de **CAC** ao invés de **CGC**, todas as ferramentas ditas “autônomas” têm potencial de serem incorporadas no processo de criação de um artista humano.

As *IAs* atualmente disponíveis ainda não conseguem substituir humanos e, mesmo quando utilizadas como ferramentas integradas a um processo de criação de um artista vivo, não conseguem realizar as tarefas com a qualidade e consistência que um humano experiente conseguiria.

8 Conclusão e direcionamentos para o futuro

Muitas perguntas mais filosóficas apresentadas na introdução continuam em aberto (conforme previsto), entretanto obtivemos algumas pistas que nos aproximam de uma compreensão sobre tais questões, seja por consultar as experiências dos artistas envolvidos com *IA*, como também por ter uma visão melhor sobre o estado atual das capacidades das tecnologias mais recentes do campo.

O crescimento relativamente recente do número de ferramentas destinadas a não especialistas da área têm colaborado com o crescimento no número de obras que se utilizam de ferramentas criadas com *IA*. Há ferramentas que realizam diversas tarefas musicais e que são desenhadas especialmente para o público artista, com interfaces intuitivas e por vezes com integração aos softwares de música (*DAWs*). As ferramentas de *IA* atraíram engajamento de artistas nas áreas de criação e derivação de ideias musicais, harmonizações, criação de instrumentos, etc. Este engajamento conta inclusive com músicos premiados e/ou de fama como Yo-Yo Ma (“*Begin Again Again...*” de Tod Machover) e *Stromae* (Músicas *Hello Shadow* e *Valise* do álbum *Hello World*).

Observamos que no cenário atual os artistas mostram uma tendência a gravitar em torno das *IAs* que funcionam como ferramentas - ou nas palavras de Robin Sloan, instrumentos musicais - que servem aos criadores durante seus processos criativos, os ajudando a criar músicas com as quais se identificam. Entretanto, ainda há ferramentas de *IAs* que possuem a capacidade de substituir humanos no processo (como a AIVA) - o que não significa que não possam ser usadas de maneira criativa por artistas. Sobre tais softwares notamos que *aparentemente* já há casos onde as criações de uma *IA* são contempladas por leis de autoria³⁵.

³⁴ Uma cópia do projeto, na qual pode-se desenvolver independentemente do projeto original.

³⁵ Deduzimos após constatar o sistema de licenças da AIVA, sem consultar nenhum documento legal. Entretanto, tal assunto merece pesquisas separadas na área de legislação.

Durante nossos experimentos com *Carmim* encontramos pontos onde a Magenta tem performance consistente (geralmente nas tarefas de derivação), porém julgamos que as habilidades de criação direta ainda são mais difíceis de serem utilizadas (pelo menos sem uma personalização dos modelos com ideias de referência). Apesar disto, o grande poder generativo das redes serve para mitigar este problema: pode-se com a facilidade de poucos cliques obter dezenas de ideias diferentes dentre as quais basta escolher as que interessam. Nas últimas sessões de *playtesting*, o jogo começou a se tornar cativante e as possibilidades pareciam bastante atraentes. Todavia, concluímos que modificações futuras ainda serão necessárias para que o jogo atinja seu verdadeiro potencial, como a implementação de um modo de criação livre além da implementação de outras funcionalidades da biblioteca *magenta.js*.

Inteligências Artificiais *cutting-edge* como a *Magenta* e *FlowMachines* se mostraram capazes de desempenhar um papel crucial na criação de álbuns por artistas de sucesso (por exemplo, como vimos, todas as melodias do álbum *Chain Tripping* foram geradas pela Magenta!). Entretanto, ainda é necessária uma “supervisão humana” e tais ferramentas são utilizadas como se fossem um instrumento musical pelos artistas. Não faremos juízo estético sobre as músicas criadas por humanos utilizando tais ferramentas, porém convidamos o leitor a ouvi-las e tirar suas próprias conclusões.

Por fim, esperamos ter proporcionado uma introdução técnica simplificada para que o leitor não especialista consiga ter uma visão simplificada de como redes neurais funcionam e como a *Magenta* realiza as operações musicais implementadas no jogo *Carmim*.

Referências

AIVA: homepage do projeto na web. Disponível em: <<https://www.aiva.ai/>>. Acesso em: 23 dez. 2021.

BHARUCHA, Jamshed. MUSACT: A connectionist model of musical harmony. In: MACHINE Models of Music. 1. ed. Cambridge, Massachusetts: MIT Press, 1993. p. 497–509. ISBN 0-262-19319-1. Disponível em: <<https://mitpress.mit.edu/books/machine-models-music>>. Acesso em: 23 mar. 2021.

BUCK, Alex Kantorowicz. **Fantasia essata: o computador a serviço da música**. 2018. f. 166. Dissertação (Mestrado) - Curso de Música: Composição – Instituto de Artes da Unesp, São Paulo. Disponível em: <<https://repositorio.unesp.br/handle/11449/157503>>.

COLLINS, Nick. Automatic Composition of Electroacoustic Art Music Utilizing Machine Listening. **Computer Music Journal**, The MIT Press, v. 36, n. 3, p. 8–23, 2012. ISSN 01489267, 15315169. Disponível em: <<http://www.jstor.org/stable/23254379>>.

CONT, Arshia. ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters in Computer Music. In: INTERNATIONAL COMPUTER MUSIC CONFERENCE (ICMC). Belfast, Ireland: [s.n.], ago. 2008. p. 33–40. Disponível em: <https://hal.inria.fr/hal-00694803>.

DORFER, Matthias; ARZT, Andreas; WIDMER, Gerhard. Towards Score Following in Sheet Music Images. **CoRR**, abs/1612.05050, 2016. arXiv: 1612.05050. Disponível em: <http://arxiv.org/abs/1612.05050>.

EUCLIDEAN Space. In: WIKIPÉDIA: a enciclopédia livre. [S.l.]: Wikimedia, 2021. Disponível em: https://en.wikipedia.org/wiki/Euclidean_space. Acesso em: 10 nov. 2021.

FORTIN, Sylvie; GOSSELIN, Pierre. Considerações metodológicas para a pesquisa em arte no meio acadêmico. **ARJ – Art Research Journal: Revista de Pesquisa em Artes**, v. 1, n. 1, p. 1–17, mai. 2014. DOI: 10.36025/arj.v1i1.5256. Disponível em: <https://periodicos.ufrn.br/artresearchjournal/article/view/5256>.

GILLICK, Jon et al. **Learning to Groove with Inverse Sequence Transformations**. [S.l.: s.n.], 2019. arXiv: 1905.06118 [cs.SD].

HELLO World: website oficial. Disponível em: <https://www.helloworldalbum.net/>. Acesso em: 26 dez. 2021.

HUANG, Cheng-Zhi Anna et al. Counterpoint by Convolution. In: INTERNATIONAL Society for Music Information Retrieval (ISMIR). [S.l.: s.n.], 2017.

INTERPOLAÇÃO Linear. In: WIKIPÉDIA: a enciclopédia livre. [S.l.]: Wikimedia, 2021. Disponível em: https://pt.wikipedia.org/wiki/Interpola%C3%A7%C3%A3o_linear. Acesso em: 5 jan. 2022.

JOYCE, Caneel. The Blank Page: Effects of Constraint on Creativity. **SSRN Electronic Journal**, dez. 2009. DOI: 10.2139/ssrn.1552835.

LECUN, Yann; CORTES, Corinna; BURGESS, Christopher J.C. **The MNIST database of handwritten digits**. 1998. Disponível em: <http://yann.lecun.com/exdb/mnist/>. Acesso em: 4 nov. 2021.

MACHINE Learning. In: WIKIPÉDIA: a enciclopédia livre. [S.l.]: Wikimedia, 2021. Disponível em: https://en.wikipedia.org/wiki/Machine_learning. Acesso em: 10 nov. 2021.

MAGENTA: homepage do projeto. Disponível em: <https://magenta.tensorflow.org/>. Acesso em: 23 dez. 2021.

MÁNTARAS, Ramón López de. Artificial Intelligence and the Arts: Toward Computational Creativity. In: **The Next Step: Exponential Life**. Edição: Turner Libros. Berlin: BBVA, 2016. p. 99–123. ISBN 978-84-16714-45-2. Disponível em: <<https://www.bbvaopenmind.com/en/books/the-next-step-exponential-life/>>. Acesso em: 18 fev. 2020.

MATTISE, Nathan. **How YACHT fed their old music to the machine and got a killer new album**. Ago. 2019. Disponível em: <<https://arstechnica.com/gaming/2019/08/yachts-chain-tripping-is-a-new-landmark-for-ai-music-an-album-that-doesnt-suck/>>. Acesso em: 26 dez. 2021.

NSYNTH: página do NSynth no website do projeto Magenta. Disponível em: <<https://magenta.tensorflow.org/nsynth-instrument>>. Acesso em: 29 dez. 2021.

OLAH, Christopher. **Understanding LSTM Networks**. Disponível em: <<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>>. Acesso em: 23 dez. 2021.

PACHA, Alexander; EIDENBERGER, Horst. Towards Self-Learning Optical Music Recognition. In: 2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA). [S.l.: s.n.], 2017. p. 795–800. DOI: [10.1109/ICMLA.2017.00-60](https://doi.org/10.1109/ICMLA.2017.00-60).

PACHET, François; ROY, Pierre; CARRÉ, Benoit. **Assisted music creation with Flow Machines: towards new categories of new**. [S.l.: s.n.], 2021. arXiv: [2006.09232 \[cs.SD\]](https://arxiv.org/abs/2006.09232).

PUERTO, Oscar; THUE, David. A Model of Inter-musician Communication for Artificial Musical Intelligence. In: ICCM. [S.l.: s.n.], 2017.

ROBERTS, Adam; HAWTHORNE, Curtis; SIMON, Ian. Magenta.js: A JavaScript API for Augmenting Creativity with Deep Learning. In: JOINT Workshop on Machine Learning for Music (ICML). [S.l.: s.n.], 2018.

ROCCA, Joseph; ROCCA, Baptiste. **Understanding Variational Autoencoders (VAEs): Building, step by step, the reasoning that leads to VAEs**. 2019. Disponível em: <<https://towardsdatascience.com/understanding-variational-autoencoders-vaes-f70510919f73>>. Acesso em: 24 out. 2021.

RODA Viva: Entrevista com Yuval Harari. 2019. Disponível em: <<https://www.youtube.com/watch?v=pBQM085IxOM>>. Acesso em: 24 out. 2021.

SALEN, Katie; ZIMMERMAN, Eric. **Regras do Jogo: Fundamentos do Design de Jogos**. 1. ed. São Paulo: Blucher, 2012. 4 v.

SLOAN, Robin. **Making the Music of the Mazg**. Disponível em: <<https://www.mcdbooks.com/features/sourdough>>. Acesso em: 26 dez. 2021.

VETOR (Matemática). In: WIKIPÉDIA: a enciclopédia livre. [S.l.]: Wikimedia, 2021.

Disponível em: <[%5Curl%7Bhttps:](https://pt.wikipedia.org/wiki/Vetor_(matem%C3%A1tica))

[//pt.wikipedia.org/wiki/Vetor_\(matem%C3%A1tica\)%7D](https://pt.wikipedia.org/wiki/Vetor_(matem%C3%A1tica))>. Acesso em: 5 jan. 2022.

VINCENT, James. **Making this album with AI ‘felt like wandering in an enormous labyrinth’**. Out. 2021. Disponível em:

<<https://www.theverge.com/2021/10/28/22750337/shadow-planet-ai-robin-sloan-jesse-solomon-clark>>. Acesso em: 26 dez. 2021.

Apêndice

Excerto criado com o jogo Carmim

Track 1

Track 2

Track 3

Track 4

Track 5

This block contains the first system of a musical score, measures 1 through 6. It consists of five tracks. Tracks 1, 2, and 3 are in bass clef with a key signature of four flats (B-flat, E-flat, A-flat, D-flat) and a 4/4 time signature. Track 1 has rests for the first four measures, followed by a half note G-flat, a quarter note F, and a half note E-flat. Track 2 has a half note G-flat, a quarter note F, a half note E-flat, a quarter note D-flat, a half note C, and a half note B-flat. Track 3 has rests for the first four measures, followed by a half note G-flat, a quarter note F, and a half note E-flat. Track 4 has rests for all six measures. Track 5 is a percussion track with rests for all six measures.

7

Track 1

Track 2

Track 3

This block contains the second system of a musical score, measures 7 through 12. It consists of three tracks. Tracks 1, 2, and 3 are in bass clef with a key signature of four flats (B-flat, E-flat, A-flat, D-flat) and a 4/4 time signature. Track 1 has a half note G-flat, a quarter note F, a half note E-flat, a quarter note D-flat, a half note C, and a half note B-flat. Track 2 has a half note G-flat, a quarter note F, a half note E-flat, a quarter note D-flat, a half note C, and a half note B-flat. Track 3 has a half note G-flat, a quarter note F, a half note E-flat, a quarter note D-flat, a half note C, and a half note B-flat.

13

Track 1

Track 3

Track 5

This block contains the third system of a musical score, measures 13 through 15. It consists of three tracks. Tracks 1, 3, and 5 are in bass clef with a key signature of four flats (B-flat, E-flat, A-flat, D-flat) and a 4/4 time signature. Track 1 has a half note G-flat, a quarter note F, a half note E-flat, a quarter note D-flat, a half note C, and a half note B-flat. Track 3 has a half note G-flat, a quarter note F, a half note E-flat, a quarter note D-flat, a half note C, and a half note B-flat. Track 5 has a half note G-flat, a quarter note F, a half note E-flat, a quarter note D-flat, a half note C, and a half note B-flat.

19

Track 1

Track 2

Track 3

Track 5

23

Track 1

Track 2

Track 3

Track 4

Track 5

Detailed description: This image shows a musical score for five tracks across three measures. The key signature is three flats (B-flat, E-flat, A-flat). Track 1 and 2 are in bass clef. Track 3 is in bass clef. Track 4 is in bass clef. Track 5 is in a percussion clef. Measure 23: Track 1 has a half note G2, a quarter note F2, a quarter note E2, and a quarter rest. Track 2 has a half note G2, a quarter note F2, a quarter note E2, and a quarter rest. Track 3 has a half note G2, a quarter rest, and a quarter rest. Track 4 has a whole rest. Track 5 has a quarter rest, followed by eighth notes G2, F2, E2, D2, C2, B1, A1, G1. Measure 24: Track 1 has a half note G2, a quarter note F2, a quarter note E2, and a quarter rest. Track 2 has a half note G2, a quarter note F2, a quarter note E2, and a quarter rest. Track 3 has a half note G2, a quarter rest, and a quarter rest. Track 4 has a whole rest. Track 5 has a quarter rest, followed by eighth notes G2, F2, E2, D2, C2, B1, A1, G1. Measure 25: Track 1 has a half note G2, a quarter note F2, a quarter note E2, and a quarter rest. Track 2 has a half note G2, a quarter note F2, a quarter note E2, and a quarter rest. Track 3 has a half note G2, a quarter note F2, a quarter note E2, and a quarter rest. Track 4 has a half note G2, a quarter note F2, a quarter note E2, and a quarter rest. Track 5 has a quarter rest, followed by eighth notes G2, F2, E2, D2, C2, B1, A1, G1.

26

Track 1

Track 2

Track 3

Track 4

Track 5

This block contains the musical notation for measures 26 through 28 across five tracks. Tracks 1, 2, and 3 are in bass clef with a key signature of three flats. Track 4 is in bass clef with a key signature of four flats. Track 5 is in a percussion clef. The notation includes various note values, rests, and articulation marks such as accents and slurs.

29

Track 1

Track 2

Track 3

Track 5

This block contains the musical notation for measures 29 and 30 across four tracks. Tracks 1 and 2 are in treble clef with a key signature of three flats. Track 3 is in bass clef with a key signature of three flats. Track 5 is in a percussion clef. The notation includes various note values, rests, and articulation marks such as accents and slurs.

31

Track 1

Track 2

Track 3

Track 5

This block contains the musical notation for measures 31 and 32 across four tracks. Tracks 1 and 2 are in treble clef with a key signature of three flats. Track 3 is in bass clef with a key signature of three flats. Track 5 is in a percussion clef. The notation includes various note values, rests, and articulation marks such as accents and slurs.

Track 5

Track 5

Track 5

39

Track 1

Track 4

Track 5

39

41

Track 1

Track 4

Track 5

41

43

Track 1

Track 4

Track 5

43

45

Track 1

Track 4

Track 5

45

47

Track 1

Track 4

Track 5

48

49

Track 1

Track 4

Track 5

50

51

Track 1

Track 4

Track 5

52

53

Track 1

Track 4

Track 5

54

55

Track 1

Track 4

Track 5

This system contains measures 55 and 56. Track 1 (treble clef) features a melodic line with eighth and sixteenth notes, including a triplet in measure 55. Track 4 (bass clef) provides a harmonic accompaniment with eighth and sixteenth notes. Track 5 (bass clef) shows a drum part with a double bar line at the start of measure 55, followed by a steady eighth-note pattern. Vertical asterisks mark the downbeats of measures 55 and 56.

57

Track 1

Track 4

Track 5

This system contains measures 57 and 58. Track 1 continues the melodic line. Track 4 continues the harmonic accompaniment. Track 5 continues the drum pattern. Vertical asterisks mark the downbeats of measures 57 and 58.

59

Track 1

Track 4

Track 5

This system contains measures 59 and 60. Track 1 concludes the melodic phrase. Track 4 concludes the harmonic accompaniment. Track 5 concludes the drum pattern. Vertical asterisks mark the downbeats of measures 59 and 60.