http://journal.unoeste.br/index.php/ca/index
DOI: 10.5747/ce.2019.v11.n2.e275
ISSN on-line 2178-8332

Colloquium Exactarum

Submetido: 16/11/2018 Revisado: 20/02/2019 Aceito: 28/06/2019

GERAÇÃO DE MÚSICA COM APRENDIZADO DE MÁQUINA

MUSIC GENERATION WITH MACHINE LEARNING

Joel Alexandre de Sá Júnior[1]; Mário Augusto Pazoti[1]; Leandro Luiz de Almeida[1]; Francisco Assis da Silva[1]; Danillo Roberto Pereira[1]

[1]Universidade do Oeste Paulista - UNOESTE
Faculdade de Informática - FIPP
Presidente Prudente, São Paulo, Brasil.
mario@unoeste.br

RESUMO – Aprendizado de máquina é um conceito que tem sido parte do dia-a-dia, sendo utilizado em aplicações como redes sociais, comércio digital, assistentes em telefones celulares, entre outras. Na área da música ele pode ser utilizado para inspirar compositores, produzir música inspirada em um estilo específico, ou gerar música em tempo real para jogos ou aplicações de RV. Este artigo irá avaliar a habilidade de uma rede neural de gerar resultados satisfatórios na área da música, utilizando a biblioteca Magenta, um conjunto de ferramentas para o uso de aprendizado de máquina em aplicações artísticas. As músicas são geradas baseadas em um dataset de composições de Bach e verificadas proceduralmente para plágio, comparando os resultados obtidos com o dataset.
Palavras-chave: Aprendizado de máquina, Música, Magenta, Redes Neurais, LSTM.

ABSTRACT – Machine learning is a concept that has been a part of the day-to-day life, being used in applications like social networks, e-commerce, smartphone assistants, among others. In the music area it can be used to inspire composers, produce music based on a specific style or generate music in real time for games or VR applications. This article will evaluate the ability of a neural network to generate satisfactory results in the music area, using Magenta libraries, a toolkit for the use of machine learning in artistic applications. The songs are generated based on a dataset of Bach compositions and procedurally checked for plagiarism, comparing the obtained results to the dataset.
Keywords: Machine learning, Music, Magenta, Neural networks, LSTM.

1 INTRODUÇÃO

O aprendizado de máquina é uma tecnologia que está presente em inúmeras aplicações utilizadas diariamente como nas redes sociais, sites de vendas, nos assistentes de smartphones, entre diversas outras aplicações, tanto comerciais quanto acadêmicas, com o objetivo de facilitar o trabalho do humano, realizando tarefas que normalmente requereriam algum tipo de conhecimento.

Aprendizado de máquina é definido como um "campo de estudo que dá aos computadores a habilidade de aprender sem serem explicitamente programados." (SAMUEL, 1959). Em outras palavras, é o estudo e criação de algoritmos que dão ao computador a capacidade de aprender com seus erros e, a partir deles, realizar previsões sobre os dados, gerando assim conjuntos de dados completamente novos.

Uma das áreas na qual o aprendizado de máquina pode contribuir é a música. O processo de composição de uma melodia pode ser demorado, além de demandar conhecimento de teoria musical e, também, algum tipo de inspiração. O aprendizado de máquina pode ajudar a reduzir esses três problemas.

Com uma máquina treinada para gerar músicas baseadas no estilo de um artista seria possível usar essa saída como inspiração para a criação novas melodias, aumentando a produtividade de um músico. Além disso, não seria necessário tanto conhecimento em teoria musical, pois, se os resultados do treinamento forem bons o bastante, a máquina teria a capacidade de gerar escalas e harmonias complexas e musicalmente coerentes.

Para alcançar resultados com esse nível de qualidade é necessária uma estrutura de rede neural que consiga visualizar a música como um todo, portanto precisa de um tipo de memória. Para isso, uma RNN, ou rede neural recorrente, se torna uma excelente alternativa.

RNNs são um tipo específico de rede neural que tem a capacidade de predizer dados futuros a partir de exemplos anteriores, e são muito poderosas para modelagem de dados sequenciais como texto ou áudio (DENG; YU, 2014).

O conceito de utilizar RNNs para produzir música já vem sendo estudado e tem gerado resultados interessantes. Chen e Miikkulainen (2001) abordaram a música como uma linguagem que segue uma "gramática musical", que é um conjunto de regras que descreve o estilo de um compositor. Foi utilizada uma SRN, ou Rede recorrente simples para gerar músicas com base no compositor Béla Bartók. Para a avaliação dos resultados, as músicas geradas foram testadas em relação ao conjunto de regras do estilo de Bartók. O objetivo era que as novas músicas fossem fieis às regras do compositor, mas com variações criativas (CHEN; MIIKKULAINEN, 2001).

Huang e Wu (2016) fizeram um estudo para a geração de música utilizando Deep Learning, que se baseia em RNNs de múltiplas camadas. O objetivo de seu trabalho era gerar músicas que não tivessem somente a melodia, mas também a harmonia agradável aos ouvidos humanos. Para facilitar o trabalho com as notas, foram utilizados dois formatos de dados. O primeiro era chamado "Midi data", no qual foram extraídas, do arquivo midi, informações de tom e duração de cada nota, gerando tokens com para a base de dados. O outro é "Piano roll" em que as notas eram tratadas como uma linha do tempo, e cada token indicavam um conjunto de notas tocadas naquele momento. O modelo utilizado foi de LSTM (Long Short Term Memory, uma variação da RNN para que consiga armazenar informações de longo prazo) de duas camadas. Para avaliar os resultados, foi realizado um teste às cegas com 26 voluntários que ouviram 3 exemplos gerados pela rede e deram notas de 1 a 10 para as músicas.(HUANG; WU, 2016)

Diante disso, definiu-se como objetivo deste trabalho a avaliação da capacidade que uma rede neural tem de produzir músicas similares às de um compositor real, mas sem

gerar cópias das músicas originais, o que as caracterizaria como plágio. Para isso, foram realizados testes entre as músicas geradas pela rede e a base de dados.

Para o treinamento e geração de músicas neste trabalho foi usada a Polyphony RNN, um modelo de rede neural do Projeto Magenta, que é um conjunto de ferramentas criadas para serem utilizadas em conjunto com o TensorFlow para trabalhar com arte.(WAITE et al.2016) Esse modelo foi escolhido, pois tem a capacidade de gerar músicas polifônicas assim como as utilizadas como base de treinamento.

A rede foi treinada utilizando um conjunto de composições de Johann Sebastian Bach, que é um compositor muito famoso do qual foi possível encontrar uma quantidade considerável de arquivos midi de suas músicas.

Os resultados foram avaliados quanto sua semelhança com a base de dados e sua qualidade musical. Para verificar se as músicas geradas pela máquina são originais com relação à base de aprendizado os arquivos midi gerados foram convertidos para CSV, facilitando seu manuseio. Com os arquivos nesse formato será possível realizar testes e comparações para garantir a originalidade das músicas. Para verificar a qualidade musical dos resultados foram realizados testes às cegas com voluntários para avaliarem se as músicas mostradas para eles são geradas por máquina ou compostas por um músico real.

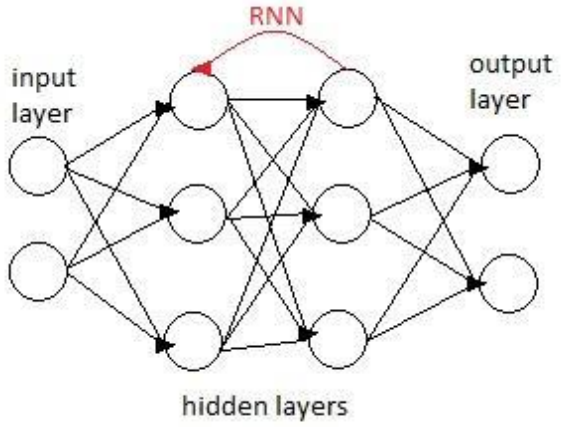
A seguir encontra-se a seção II, de materiais e métodos onde serão explicados mais profundamente os conceitos utilizados neste trabalho como RNN, LSTM, Polyphony RNN e os testes realizados sobre as músicas. A seção III fala sobre os resultados obtidos ao fim do treinamento e uma discussão aprofundada sobre os resultados das avaliações, tanto de semelhança quanto de qualidade e o artigo é finalizado com a seção IV que é a conclusão com agradecimentos e bibliografia.

2 MATERIAIS E MÉTODOS

2.1 Redes Neurais Recorrentes (RNN)

RNNs são uma classe de redes profundas que podem ser usadas tanto em aprendizado supervisionado ou não, e tem a capacidade de predizer os dados do futuro a partir de exemplos anteriores. RNNs são muito poderosas para modelagem de dados sequenciais como texto ou áudio (DENG; YU, 2014). Como os dados nesse estudo serão formatados em texto, como pode ser visto na seção 3, esse modelo será muito útil. Na Fig. 1 pode ser visto um exemplo de RNN.

Figura 1. Exemplo de RNN.



Fonte: O autor.

Uma boa maneira de explicar RNN é que elas tem um tipo de memória, que auxilia a realização de decisões futuras a partir de aprendizado passado.

Em conjunto com o conceito de RNN, sempre se fala da Deep Learning, ou aprendizado profundo, que é "uma classe de técnicas de aprendizado de máquina que explora várias camadas de informações não lineares para extração de aspectos supervisionada ou não supervisionada e para análise de padrões e classificação”(DENG; YU, 2014). A criação dessa área se deu pelo entendimento melhor do funcionamento do Neocórtex, uma região do cérebro associada a habilidades cognitivas, que realiza a propagação dos pulsos elétricos através de uma hierarquia

complexa para aprender com as regularidades apresentadas. (AREL;KARNOWSKI, 2010)

Um problema que encontramos nas RNNs é que elas "não podem acompanhar eventos temporariamente distantes que indicam estrutura de música global"(ECK; SCHMIDHUBER, 2002) , por isso pode ser melhor a utilização de uma LSTM, que é um tipo específico de RNN que tem "unidades escondidas que tem o comportamento natural de guardar dados de entrada por um longo tempo" (LECUN; BENGIO; HINTON, 2015).

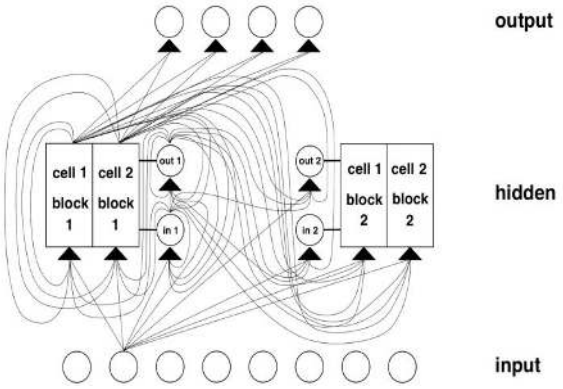
2.2 Long-Short Term Memory (LSTM)

LSTMs são um grupo ainda mais específico de redes neurais. Apesar das Redes Neurais Recorrentes terem a capacidade de utilizar informações antigas para tomar decisões, elas somente podem "lembrar" de informações recentes. Como uma música precisa ser fiel a diversas estruturas por toda sua extensão, foi necessário encontrar um modelo que mantivesse em sua memória informações de longo prazo, e essa é uma característica chave das LSTMS.

A "memória" das RNNs são conexões de feedback utilizadas para armazenar informações de entradas anteriores. O problema é que conforme essas conexões são utilizadas e os erros fluem de maneira reversa na rede, podem ocorrer dois problemas (1) o acúmulo de erros, até uma explosão ou (2) desaparecimento, pois o erro retropropagado depende exponencialmente dos pesos.

A solução encontrada foi uma nova estrutura, que faz o uso de um fluxo de erro constante através de unidades especiais chamadas células (cells) dentro da rede (HORCHEITER; SCHMIDHUBER, 1997), como pode ser visto na Fig. 2. As células armazenam a entrada da camada anterior e a entrada atual e escolhem o que será mantido ou apagado. Então são combinados o estado anterior, a memória atual a entrada atual, capturando assim dependências de longo prazo.

Figura 2. Estrutura de uma LSTM. (Horcheiter e Schmidhuber, 1997)



Fonte: Hochreiter e Schmidhuber (1997)

2.3 TensorFlow

TensorFlow é uma biblioteca de aplicações de aprendizado de máquina da Google, que dá a possibilidade de serem utilizados diversos algoritmos de aprendizado e otimizações para o treinamento. Além disso, tem o foco na criação de redes neurais profundas, que serão utilizadas nesse trabalho, além de ter uma grande comunidade e recursos para o suporte.

2.4 Magenta

Magenta é um projeto da Google baseado no TensorFlow, sua biblioteca para aplicações de aprendizado de máquina, que envolve pesquisas e ferramentas para o uso de aprendizado de máquina em aplicações artísticas como música, desenho e pintura.[5]

A equipe de pesquisadores do Magenta disponibilizou um conjunto de modelos que podem ser utilizados para treinar redes neurais com propriedades específicas. Neste estudo será utilizado a Polyphony RNN.

O modelo Polyphony RNN aplica modelagem de linguagem para música utilizando uma LSTM, abordada anteriormente. É esperado que esse modelo possa processar e gerar resultados que contenham a reprodução de múltiplas notas ao mesmo tempo. A escolha desse modelo foi trivial tendo em vista o conjunto de dados de entradas, que são músicas completas de piano. Nela as notas são tratadas

em conjuntos referentes aos compassos. Um compasso é denotado pelos símbolos START e END. Cada tempo de um compasso tem o seu final marcado pelo símbolo STEP END. As notas são indicadas por NEW_NOTE, <N> quando são tocadas e CONTINUED_NOTE <N> quando são estendidas. N é o número referente à nota no rolo de piano. Na Fig. 3 podemos ver um acorde de dó segurado por 4 tempos.

Figura 3. Estrutura utilizada na Polyphony RNN.

```
START
NEW_NOTE, 67
NEW_NOTE, 64
NEW_NOTE, 60
STEP_END
CONTINUED_NOTE, 67
CONTINUED_NOTE, 64
CONTINUED_NOTE, 60
STEP_END
CONTINUED_NOTE, 67
CONTINUED_NOTE, 64
CONTINUED_NOTE, 60
STEP_END
CONTINUED_NOTE, 67
CONTINUED_NOTE, 64
CONTINUED_NOTE, 60
STEP_END
END
```

Fonte:O autor.

2.4 Representação dos dados

No campo da produção musical um formato de dados muito utilizado é o MIDI, que representa explicitamente as notas, sua intensidade, sua duração e tempo. Esse conjunto de informações é perfeito para o objetivo deste trabalho. Com eles será possível obter características detalhadas para o aprendizado.

Como serão utilizadas as ferramentas do Magenta para realizar o treinamento da rede, esses arquivos midi serão convertidos por um formato chamado NoteSequences, que dá ao TensorFlow uma melhor capacidade para processar os dados. Durante o treinamento é possível exportar arquivos MIDI diretamente, facilitando a reprodução e pós-processamento das músicas.

Os arquivos MIDI também podem ser convertidos em um formato textual como pode ser visto na Figura 3. Isso dá recursos para o pós processamento dos dados, fazendo testes estruturais da música e também realizar as verificações de plágio nas novas músicas.

2.5 Base de aprendizado

É possível encontrar inúmeros arquivos midi disponíveis para download na internet, de diversos compositores. Todos esses arquivos poderão ser utilizados, porém é interessante escolher um compositor específico para que se possa verificar se o sistema realizou a simulação de maneira correta características do mesmo. Por questões de preferência, foram escolhidas músicas de Bach como base de aprendizado para a rede neural. Será realizado um estudo de características estilísticas dele para usar como uma verificação final dos resultados.

2.6 Treinamento

Foram realizados três testes de treinamento da rede neural, em três máquinas diferentes. O primeiro foi um teste preliminar, somente para entender a instalação e funcionamento da do Magenta. Esse teste foi realizado em um Macbook Pro com 8 GB de memória RAM, e um processador Intel Core i5, com 2 núcleos e 2,4 GHz. Esse treinamento não apresentou resultados relevantes por ser uma máquina muito inferior ao esperado para um treinamento de qualidade.

Posteriormente foi realizado o treinamento em uma máquina que possui GPU. Isso possibilitou um aumento considerável na velocidade de treinamento. As configurações dessa máquina são 8 GB de memória RAM, processador Intel Core i5 de 8 núcleos e 3.8 GHz, e uma placa de vídeo Nvidia GeForce GTX 1060 com 6 GB de VRAM. O sistema operacional utilizado foi Ubuntu 18.04 LTS.

O último teste foi realizado utilizando um recurso de Cloud Computing da Google, para execução remota do treinamento. Para isso foi criada uma instancia de Ubuntu 18.04 LTS com processador 16 núcleos Intel Haswell e 60 GB de memória RAM. Esse teste foi o que obteve os resultados mais satisfatórios, alcançando o menor valor de perda ao fim da execução.

Nos dois primeiros testes a rede neural foi configurada com duas camadas, cada uma com 64 nós. Com essas configurações foi possível perceber que os passos de treinamento ou gerações passavam rapidamente, mas a queda do valor de perda era consideravelmente mais lenta, no caso do teste com GPU, chegando ao final das 20000 gerações em cerca de 15 horas com aproximadamente 0.8 de perda.

No teste em cloud computing foi utilizada uma rede neural com três camadas, cada uma com 128 nós. Nesse caso o treinamento foi muito mais demorado, levando 81 horas para terminar as 20000 gerações, porém ao final do treinamento alcançou o valor de 0.33 de perda.

Os arquivos utilizados no decorrer do estudo foram os gerados pela instância de cloud computing por ter obtido o menor valor de perda, gerando assim resultados musicalmente mais satisfatórios. Na Fig. 4 estão presentes os comandos utilizados para o treinamento.

Figura 4. Comandos para conversão da base de dados e treinamento da Polyphony RNN

```
INPUT_DIRECTORY=/home/bach_midi
SEQUENCES_TFRECORD=/home/notesequences.tfrecord

convert_dir_to_note_sequences \
--input_dir=$INPUT_DIRECTORY \
--output_file=$SEQUENCES_TFRECORD \
--recursive

polyphony_rnn_create_dataset \
--input=/home/jnr/notesequences.tfrecord \
--output_dir=/home/jnr/polyphony_rnn/sequence_examples \
--eval_ratio=0.10

polyphony_rnn_train \
--run_dir=/home/joel_asj2/mgnta/logdir/run1 \
--sequence_example_file=/home/joel_asj2/mgnta/sequence_examples/training_poly_tracks.tfrecord \
--hparams="batch_size=64,rnn_layer_sizes=[128,128,128]" \
--num_training_steps=20000

polyphony_rnn_generate \
--run_dir=/home/joel_asj2/mgnta/logdir/run1 \
--hparams="batch_size=64,rnn_layer_sizes=[128,128,128]" \
--output_dir=/home/joel_asj2/mgnta \
--num_outputs=10 --num_steps=128 \
--primer_pitches="[67,64,60]" \
--condition_on_primer=true \
--inject_primer_during_generation=false
```

Fonte: O autor.

2.7 Formatação dos arquivos midi

Após a obtenção dos resultados, os arquivos midi gerados pela rede precisam ser formatados de uma maneira que facilite seu manuseio e comparações entre arquivos. Para isso a tanto a base da dados quanto os resultados serão convertidos de midi para CSV usando a biblioteca de Python chamada midicsv.

Mesmo após a conversão, os arquivos ainda não são homogêneos entre si, causando assim conflitos e irregularidades para a realização de comparações entre as músicas. Para resolver esse problema foi desenvolvido um script em Python que remove todo tipo de informação irrelevante para a comparação, como informações de cabeçalhos de faixas, andamento, e dados de configuração do midi. As únicas informações deixadas nos arquivos CSV foram relacionadas às notas de cada arquivo, ordenadas com relação ao momento da música que são executadas.

2.8 Testes sobre os resultados

Foram realizados dois testes quantitativos e um teste qualitativo sobre os resultados do treinamento.

O primeiro teste quantitativo foi uma verificação que buscava dentro da base de aprendizado a música que continha mais notas em comum com cada música gerada pela rede neural. Grande parte do estilo de um compositor se baseia nas escalas que o mesmo utiliza. Então, se uma música gerada pela rede possui uma alta porcentagem de notas em comum com uma da base fica claro que ela seguiu padrões estilísticos do compositor em questão.

O segundo teste quantitativo foi realizado com base no anterior. Após encontrar o arquivo com maior semelhança de escala com uma música gerada, foi realizada uma busca da maior subsequência em comum entre as duas músicas. Nesse caso, espera-se obter os menores valores possíveis, pois uma sequência de notas em comum muito grande pode acabar caracterizando um tipo de plágio, provando que a rede neural está simplesmente copiando a base de aprendizado.

O terceiro teste não tinha o objetivo de avaliar os resultados de maneira quantitativa e exata, mas de avaliar a reação de pessoas com relação a uma música gerada por uma máquina. Foi desenvolvido um questionário que tinha dois trechos de músicas, um composto por Bach e

um gerado pela rede neural. Foi solicitado que os participantes escolhessem qual das duas eles acreditavam que tinha sido gerada por rede neural. Posteriormente foi pedida uma nota de 0 a 10 para cada música. O teste foi disponibilizado em uma rede social e foram obtidas 154 respostas nesse questionário.

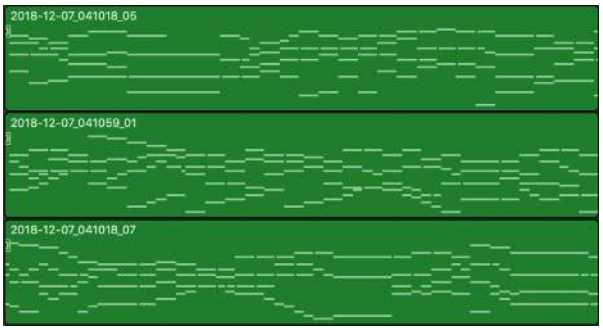
3 RESULTADOS E DISCUSSÃO

3.1 Resultados do treinamento

O treinamento que alcançou os melhores foi utilizando Cloud Computing, que após 81 horas de treinamento fez com que a rede alcançasse o valor de 0.33 de perda. Com esse valor as músicas geradas continham melodias e harmonias coerentes com músicas geradas por seres humanos. O que restava era avaliar se realmente as notas seguiam os padrões estilísticos de Bach, mas sem copiar suas melodias exatamente.

Após esse treinamento foram geradas 20 exemplos de músicas para realização dos testes comparativos com a base de dados. A Fig. 5 contém três trechos de músicas geradas pela rede neural.

Figura 5. Músicas geradas após o treinamento.



Fonte: O autor.

3.2 Testes Quantitativos

No primeiro teste esperava-se obter com qual música da base de aprendizado a melodia gerada possuía a maior semelhança estilística, e a porcentagem dessa semelhança. Um resultado com porcentagens altas nesse caso demonstra a capacidade da rede de gerar músicas estilisticamente semelhantes às oferecidas como base de aprendizado.

Os resultados obtidos nesse teste foram muito satisfatórios, pois das 20 músicas geradas pela rede neural, 17 tinham 100% das suas notas presentes em alguma música de Bach. Das 3 que restaram, a menor porcentagem de semelhança foi 98%.

O segundo teste tinha como objetivo verificar a maior sequência em comum entre as músicas que no teste anterior já foram identificadas por ter uma escala similar. Aqui se espera um resultado com valor pequeno, provando assim que as melodias não estão sendo copiadas pela rede neural.

Assim como no primeiro teste, os resultados foram muito satisfatórios, pois das 20 músicas 19 apresentaram somente 2 notas na maior sequência em comum, e uma apresentou 3 notas. Com esses números fica claro que a rede neural não está copiando as melodias originais. Uma sequência de apenas três notas dificilmente caracterizaria um plágio de uma música de Bach.

Tabela 1. Contagem de notas e sequências em comum com a base de dados.

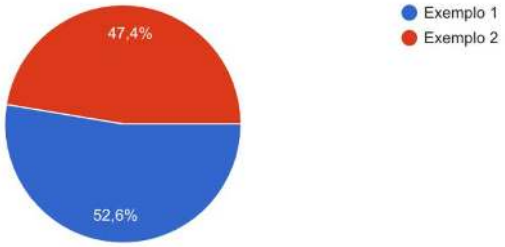
Exemplo	% das notas	Maior Sequência Comum	Música de Bach
1	100	2	033100B
2	100	2	028900B
3	100	2	027600B
4	100	2	012606B
5	100	2	024454B
6	100	2	008707B
7	100	2	027600B
8	100	2	002506B
9	98	2	038000B
10	100	2	024454B
11	100	2	024515B
12	100	2	040500B
13	100	2	024515B
14	100	2	026600B
15	100	2	032800B
16	98	3	022902B
17	99	2	040500B
18	100	2	002506B
19	100	2	027500B
20	100	2	027600B

Fonte: O autor.

3.3 Questionário

O questionário disponibilizado na internet recebeu 154 respostas. Na primeira questão foi pedido para os participantes tentarem escolher entre duas músicas qual tinha sido gerada por uma rede neural. Os resultados mostraram que 52,6% dos participantes acharam que a música composta por Bach tinha sido gerada por rede neural, como pode ser visto na Fig. 6. O questionário está disponível em <https://docs.google.com/forms/d/e/1FAIpQLSdyLFXo8xDvuoGmBrCRbH2dNQRyElVk5s8ouluIEwlaAqifxw/viewform?usp=sf_link>.

Figura 6. Resultados da primeira questão da pesquisa, onde o Exemplo 1 é a música de Bach e o exemplo 2 a gerada pela Polyphony RNN.

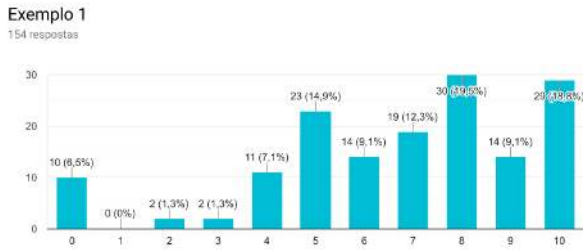


Fonte: O autor

Alguns dos participantes foram entrevistados para podermos entender o porquê de terem escolhido essa opção. Uma resposta recorrente foi que como a melodia de Bach era mais simples, tinha-se a impressão de ser artificial, enquanto a da IA era mais inventiva com suas melodias. Entretanto, alguns entrevistados com mais conhecimento em teoria musical acharam a melodia gerada por IA bagunçada e aleatória, apesar de ter melodias e harmonias coerentes, enquanto a de Bach era mais simples e controlada, como se tivesse sido composta com consciência.

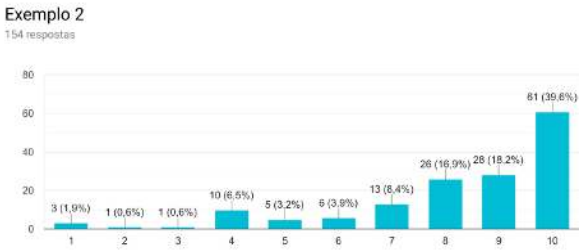
As questões de número 2 e 3 solicitaram do participante uma nota de 0 a 10 para o Exemplo 1 (Bach) e a para o Exemplo 2 (Polyphony RNN) respectivamente, ainda sem saber qual tinha sido gerada por IA, 0 indicando que a música era formada por notas completamente aleatórias e 10 que tinha sido composta por um músico profissional. Os resultados podem ser vistos na Fig. 7 para o exemplo 1 e na Fig. 8 para o exemplo 2 A nota média do Exemplo 1 foi 6.76 enquanto a nota média do Exemplo 2 foi 8.24. Mais uma vez, os participantes demonstraram preferência pela melodia mais dinâmica, por mais que fosse caótica a ouvidos mais experientes na área da música.

Figura 7. Resultados da segunda questão, que pedia uma nota de 0 a 10 para a música composta por Bach.



Fonte: O autor.

Figura 8. Resultados da terceira questão, que pedia uma nota de 0 a 10 para a música composta por Bach.



Fonte: O autor.

4 CONCLUSÕES E TRABALHOS FUTUROS

Nesse estudo conseguimos concluir que geração de música utilizando aprendizado de máquina pode produzir resultados relevantes e com qualidade, seguindo o padrão estilístico do compositor utilizado como base de aprendizado, através das escalas. Apesar disso, ficou claro que os resultados produzidos são melodias originais, sem grandes seqüências em comum com relação às músicas do compositor.

No teste às cegas realizado através do questionário podemos constatar que uma música gerada por uma inteligência artificial pode ser confundida com músicas geradas por compositores reais. Isso é mais um indicativo da qualidade alcançada pela rede neural. [11]

Com esses resultados em mãos, pode-se entender que o aprendizado de máquina aplicado na música tem a capacidade de produzir músicas com qualidade o bastante para servir como uma ferramenta para facilitar o

trabalho de compositores ou oferecer uma experiência única para diversas aplicações no mercado do entretenimento.

Em trabalhos futuros podem ser realizados testes utilizando outros modelos do projeto Magenta, como a Performance RNN, que produz melodias com dinâmica de intensidade e tempo. Além disso, pode ser desenvolvido um modelo de rede neural para ser comparado com os modelos desenvolvidos pela Google.

5 AGRADECIMENTOS

Agradecemos à Universidade do Oeste Paulista por disponibilizar o conhecimento necessário para o desenvolvimento dessa pesquisa. Também agradecemos a todos os voluntários que participaram da pesquisa através do questionário, por oferecer respostas que levaram a conclusões importantes neste trabalho.

REFERÊNCIAS

AREL, I.; ROSE, D.; KARNOWSKI, T. Deep Machine Learning - A new Frontier in Artificial Intelligence Research. **The University of Tennessee**, 2010. https://doi.org/10.1109/MCI.2010.938364

CHEN, C-C J.; MIIKKULAINEN, R. Creating melodies with evolving recurrent neural networks, In: INTERNATIONAL JOINT CONFERENCE ON NEURAL NETWORKS, 2001. Washington, DC **Proceedings** […].Washington, DC, 2001.

DENG, L.; YU, D. Deep Learning: Methods and Applications.Foundations and Trends. . **Signal Processing** v.7, n. /-3/4, p.197, 2014. https://doi.org/10.1561/2000000039

ECK, D.; SCHMIDHUBER, J. A First Look at Music Composition using LSTM Recurrent Neural Networks. **Technical Report** No. IDSIA-07-02. Istituto Dalle Molle Di Studi Sull Intelligenza Artificiale, 2002.

GOOGLE Brain, TensorFlow: A System for Large-Scale Machine Learning, **12th USENIX Symposium on Operating Systems Design and Implementation**, 2016.

HOCHREITER, S..; SCHMIDHUBER, J.Long-Short Term Memory **Neural Computation** v.9, p. 1735-1780, 1993. https://doi.org/10.1162/neco.1997.9.8.1735

HUANG, A.; WU, R. **Deep learning for music.** Universe de Stanford, 2016.

LECUN, Y.; BENGIO, Y.; HINTOM, G. Deep Learning, **Nature,** v. 521, p. 436-444, 2015. https://doi.org/10.1038/nature14539

SAMUEL, A. Some Studies in Machine Learning Using the Game of Checkers, **IBM Journal of Research and Development**, v. 3, n. 3, 1959. https://doi.org/10.1147/rd.33.0210

WAITE, E.; ECK, D.; ROBERTS, D.; ABOLAFIA, D. **Project Magenta**. 2016. Disponível em: https: //magenta.tensorflow.org. Acesso em: 18 mar. 2018.