

PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
NÚCLEO DE EDUCAÇÃO A DISTÂNCIA
Pós-graduação *Lato Sensu* em Inteligência Artificial e Aprendizado de Máquina

Andres Antonio Alvarado Rojas

Rede Neural Recorrente para Geração de Música

Natal
Junho de 2022

Andres Antonio Alvarado Rojas

REDE NEURAL RECORRENTE PARA GERAÇÃO DE MÚSICA

Trabalho de Conclusão de Curso apresentado
ao Curso de Especialização em Inteligência
Artificial e Aprendizado de Máquina, como
requisito parcial à obtenção do título de
Especialista.

Belo Horizonte

Junho de 2022

SUMÁRIO

| | |
|---|-----------|
| Introdução | 4 |
| Descrição técnica do Problema e da Solução Proposta | 4 |
| Canvas Analítico | 5 |
| Coleta de Dados | 5 |
| Processamento/Tratamento de Dados | 6 |
| Análise e Exploração dos Dados | 7 |
| Preparação dos Dados para os Modelos de Aprendizado de Máquina | 7 |
| Aplicação de Modelos de Aprendizado de Máquina | 7 |
| Discussão dos Resultados | 8 |
| Conclusão | 8 |
| Anexos | 9 |
| Anexo 1 - Canvas Analítico | 9 |
| Links | 10 |

1. Introdução

Pensando no contexto da Inteligência Artificial e Aprendizado de Máquina, nos últimos anos têm aparecido modelos que aprendem a estrutura de, por exemplo, roteiros de algum seriado específico, e são capazes de construir um roteiro “novo”, baseado no que conseguiram aprender após o treinamento com os roteiros reais.

Este tipo de aplicação, mesmo sendo feitas na sua maioria com o intuito de entretenimento, sem ter realmente uma finalidade comercial ou acadêmica, geram uma sensação de assombro, pois o modelo é capaz de criar um documento artificial, que, por muitas vezes, tem suficiente similaridade para outras pessoas os reconhecerem em relação ao tema real (de qual seriado é aquele roteiro, e.g.), mesmo sem nenhuma informação explícita, apenas pelo jeito em que foi estruturado.

Baseado nesse tipo de modelo, o objetivo deste projeto é fazer que uma rede neural aprenda e seja capaz de gerar, pelo menos em certo grau de reconhecimento, a estrutura musical de peças clássicas em formato MIDI, que seja reconhecível como "música", e não somente como um compilado de notas, ao ouvido humano.

2. Descrição técnica do Problema e da Solução Proposta

Pode se fazer um paralelo entre a estrutura textual de um documento, por exemplo, um roteiro, como exemplificado anteriormente, com a estrutura de uma partitura musical: no texto, uma sequência de palavras em sequência coerente expressam a ideia do que foi escrito; numa partitura musical, uma sequência de notas com tom e duração coerente entre elas gera uma expressão artística musical que resulta, de forma geral, prazerosa ao ouvido humano.

Por outro lado, no contexto de música digital, um dos padrões mais utilizados atualmente é o padrão MIDI (acrônimo do inglês Musical Instrument Digital Interface - Interface Digital de Instrumentos Musicais). O MIDI carrega mensagens que especificam notação, pitch (altura), velocidade, vibrato, panning, pistas e sinais de MIDI clock, que sincroniza o ritmo entre vários dispositivos. As mensagens de Voz

do Canal transmitem dados de desempenho em tempo real sobre um único canal. Mensagens "note-on" contém um número de nota MIDI que especifica o tom da nota, um valor de velocidade que indica com que força a nota foi tocada, e o número do canal; mensagens "note-off" finalizam uma nota.

Assim, este trabalho tem como objetivo treinar uma Rede Neural Recorrente, escolhida devido à natureza sequencial das notas musicais, com um conjunto de dados de arquivos musicais MIDI de duas faixas (mãos direita e esquerda de um piano) tratados a serem mais facilmente interpretados pelo modelo, e posteriormente, a geração automática de um arquivo MIDI novo, que faça certo sentido musical.

3. Canvas Analítico

Ver [Anexo 1](#).

4. Coleta de Dados

Os dados de treinamento foram retirados do site [Mutopia Project](#), que dispõe de um repositório de arquivos musicais clássicos, para uso livre.

Foram filtrados os arquivos pelos arquivos que tivessem o piano como um dos seus instrumentos, e, a partir da primeira página do filtro, foi usado um Spider (crawler do módulo Scrapy para Python) para a coleta das informações, download dos arquivos MIDI de cada entrada e o tracking dessas informações em um arquivo CSV. O arquivo contendo o Jupyter notebook do script pode ser encontrado [aqui](#). A versão completa dos CSV pode ser encontrada [aqui](#).

Para o Spider foi necessária a configuração de um Pipeline customizado baseado no FilesPipeline base do módulo Scrapy.

O crawl coleta as informações de todas as entradas da página que está sendo acessada, após isso, faz a busca pelo link de "Next 10" correspondente à próxima página de resultados e a acessa para fazer o crawl dela, até a última página (aquela que não tem mais link de próxima página).

5. Processamento/Tratamento de Dados

Após o download dos arquivos, um pré processamento foi feito nos arquivos para serem mais facilmente entendidos na fase de treinamento da rede neural. O módulo python [pretty_midi](#) foi utilizado para o gerenciamento dos arquivos, tanto para verificar atributos como para geração dos dados tratados.

Foi verificado que muitos dos arquivos continham várias faixas de instrumentos diferentes, então o primeiro passo foi o de filtrar apenas as faixas de instrumentos que estivessem nas classes de “Piano” ou “Organ”, pelas definições do protocolo MIDI ([mais informações](#)). Mesmo que na descrição do site sobre os arquivos estivesse informado que todas tinham “piano” como um dos seus instrumentos, uns poucos arquivos foram removidos durante essa fase do pré-processamento, pois suas faixas de instrumentos não cumpriram os requisitos explicados anteriormente.

Posteriormente, as faixas de instrumentos filtradas, foram “mergeadas” em uma única faixa, com o objetivo de facilitar a forma em que os dados serão colocados como entrada durante o processo de treinamento do modelo RNN.

Um ponto adicional foi a padronização de atributo de “*velocidade*” para todas as notas dos arquivos tratados para um valor arbitrário de 60. Este atributo é referente ao volume, ou força com o qual a tecla é pressionada, no caso de um piano, pelo qual este atributo foi considerado não relevante para o aprendizado do modelo, já que não influi na estrutura musical, os valores preservados foram fundamentalmente aqueles de tom e duração da nota.

Esses arquivos tratados foram salvos em uma base separada àquela dos arquivos baixados do site.

A implementação do pré-processamento de dados detalhado nesta seção pode ser encontrada [aqui](#).

6. Análise e Exploração dos Dados

Nessa etapa você começará a explorar seus dados de uma forma mais analítica, tentando elaborar ideias, levantar hipóteses e começando a identificar padrões em seus dados. Talvez você sinta a necessidade de voltar em passos anteriores, obter mais dados e tratá-los para conseguir responder ao problema proposto. Use e abuse de ferramentas estatísticas consistentes como testes de hipóteses, intervalos de confiança. Plote gráficos que te ajudem a obter insights interessantes: desde os mais simples até gráficos mais sofisticados como boxplots, mapas de calor, etc. Aqui o uso do Python e/ou R e suas poderosas bibliotecas gráficas (Matplotlib, Seaborn, ggPlot2, etc). Apresente trechos de código com as devidas justificativas.

7. Preparação dos Dados para os Modelos de Aprendizado de Máquina

Nesta etapa você deve descrever os tratamentos realizados especificamente para os modelos de Aprendizado de Máquina escolhidos, como por exemplo a criação de atributos, o balanceamento da base de dados (*undersampling* ou *oversampling*), divisão da base em treino, validação e teste, entre outros.

8. Aplicação de Modelos de Aprendizado de Máquina

Nesta seção você deve apresentar os modelos de Aprendizado de Máquina desenvolvidos no trabalho. Mostre partes do código-fonte para ilustrar a implementação de cada modelo, além do pipeline completo do processo. A escolha dos modelos deve ser adequada ao problema proposto. Embora possa ser considerado o uso de ferramentas como Weka, Knime e Orange, por exemplo, encoraja-se a implementação com linguagens como Python ou R. É importante testar mais de um tipo de algoritmo, para que resultados distintos possam ser comparados. Por exemplo, se o trabalho trata de uma classificação, modelos como Árvores de Decisão, Redes Neurais Artificiais e Support Vector Machine poderiam

ser utilizados. Além disso, devem ser escolhidas e implementadas as métricas adequadas ao problema proposto, bem como os seus resultados apresentados.

9. Discussão dos Resultados






Nesta seção você deve relatar os resultados alcançados ao final do trabalho. Mostre os resultados das métricas adotadas, seja através de gráficos, tabelas, dentre outros, que permitam a validação do seu trabalho.

10. Conclusão

Nesta seção você deve apresentar um fechamento para o trabalho. É importante apresentar um breve resumo do trabalho, resgatando o problema, como foi tratado e os resultados obtidos, bem como as limitações e perspectivas (trabalhos futuros).

Software Analytics Canvas

Project: Rede Neural Recorrente para Geração de Música

| | | | |
|--|---|---|--|
|  1. Question <i>What is it that we want to know about the software / processes / usage / organization / etc.?</i> |  2. Data Sources <i>Which data can possibly answer our question? What information do we need?</i> |  3. Heuristics <i>Which assumptions do we want to make to simplify the answer to our question?</i> |  4. Validation <i>What results do we expect from our analysis, how are they reviewed and presented in an understandable way?</i> |
| <p>Analisar a possibilidade de uma rede neural recorrente ser treinada com arquivos de áudio MIDI com o objetivo de ser capaz de gerar novos arquivos de áudio gerados artificialmente.</p> | <p>Os dados para o treinamento da rede seriam arquivos MIDI vindos de sites de compartilhamento gratuito de mídia MIDI.</p> <p>Já a análise dos resultados, dos arquivos MIDI gerados artificialmente, não faz sentido comparar os arquivos contra alguma base de verificação, pois não se tem definido objetivamente como deveria ser o resultado: a verificação seria subjetiva, já que não é trivial separar o que tenha um "sentido musical" do que não.</p> | <p>Para a simplificação das tarefas de treinamento, serão escolhidos arquivos MIDI de duas faixas, preferencialmente que sejam entendidos como uma faixa de mão direita e uma de mão esquerda de um piano (melodia e harmonia ou contrame-lodia).</p> | <p>O resultado esperado é o da geração de arquivos MIDI artificiais que sejam reconhecíveis como tendo senso musical, mesmo não necessariamente seguindo padrões convencionais de construção melódica ou de métrica. Os arquivos artificiais podem ser misturados com arquivos reais, e um grupo focal de pessoas tentar categorizar os áudios como reais ou gerados pelo modelo, para verificar se os artificiais são similares o suficiente aos reais.</p> |
|  5. Implementation <i>How can we implement the analysis step by step and in a comprehensible way?</i> | <p>A partir do banco de dados dos arquivos MIDI reais, processar eles para serem melhor "entendidos" pelo modelo na hora do treinamento. Treinar o modelo RNN com esses dados, e, depois de finalizado o treinamento, gerar arquivos MIDI artificiais.</p> <p>Para a verificação, mesmo que subjetiva, ser mais abrangente, juntar um grupo focal de pessoas aleatórias e mostrar um grupo de arquivos onde alguns serão os gerados artificialmente e outros arquivos da base real (que não sejam músicas tão conhecidas pelo público geral) e pedir para categorizarem os arquivos como "Gerados pela IA" e "Gerado por Humano", e analisar a pesquisa para verificar o quão similar a sonoridade dos arquivos artificiais parece à dos reais, com base na opinião do grupo focal.</p> | <p>What are the main insights from our analysis?</p> <p>Analisar a capacidade do modelo de criar arquivos MIDI suficiente - mente parecidos, estruturalmente, a um arquivo real.</p> | <p>What follow-up actions can we derive from the findings? Who or what do we need to address next?</p> <p>Próximos passos poderiam consistir em aumentar a quantidade de faixas que o modelo é capaz de aprender, separar bases de treinamento em categorias (música clássica romântica, blues, rock...) com o objetivo de verificar se o modelo consegue aprender as características de cada estilo musical.</p> |

11. Anexos

a. Anexo 1 - Canvas Analítico

12. Links

Repositorio Github com o Projeto: <https://github.com/andresaar/PUC-Projeto>