Ashlyn Averett

Dr. Khan

Independent Study/SRACE Creative Endeavor

4/30/2021

                                             Creative Endeavor with AI and Dance

Abstract:

This study investigates using Artificial Intelligence to create a dance sequence. The project starts with the beginning idea to create a great sequence based on rules. The rules are a preference based on your dance background, but they let you know what kind of dance steps cannot be next to each other. This project uses a genetic algorithm to find the fittest sequence of steps and later pushes it to a larger sequence. This project is split into 3 parts: the genetic algorithm, connecting the genetic algorithm with the data set, and creating an Anvil interface that pulls the python code. The result of this project is an application that takes a user input and based on the number of dance sets, they will find the best combination of dance steps. These dance steps can then be used in a dance choreography.

Introduction:

I was three years old when my mom enrolled me in dance classes. At the age of eight, I was given the choice of continuing my dance classes or other activities, I chose dance. At nine, I made the dance competition team, from that point on all free time outside of school was devoted to dance. With seventeen years of dance experience in various genres it was natural to become a part of the dance minor program. Dance Composition class was very eye-opening and difficult. Dance composition class explores compositional themes of dance through movement and choreography. Being a Computer science major, I decided to explore combining techniques learned in Dance composition with coding. Combining these two areas of studies will push me past obstacles known as “choreography block.”

This project aimed to collect dance motion capture data to use as input data for machine learning and to supervise a cognitive architecture in creating a jazz dance. I use a Noitom Perception Neuron device to capture a variety of American Jazz Dance steps. This project is focused on the data collection and an application that will use artificial intelligence and cognitive modeling to computationally generate unique American Jazz Choreography. I combined my dance minor and computer science major to conduct an ongoing project to create an organized database of motion capture data for use in artificial intelligence research to generate computational creativity. I am currently using a genetic algorithm to find the best sequence of steps and then concatenate them into a list.

Background:

Jazz dance was based upon the Pattin’ juba a social dance brought by Africans to Americans during the transatlantic slave trade. It featured elements we now recognize as jazz dance, like improvisation, exploration of rhythm, and playfulness. Jazz has also laid the foundation of tap, Broadway musical theatre, and hip hop. Throughout the 1940’s and 50’s jazz became codified, giving us the techniques known worldwide from people like Matt Mattox, Luigi, Gus Giordano and Bob Fosse (Nicholls).

In our jazz class, our professor Brenda Hamilton expressed to us that dance steps could be separated into 4 categories, floorwork, footwork, leaps and jumps, and turns. I built off this idea plus the fact that a step or walk is needed between dance steps and came up with 5 categories.

Genetic Algorithms are a part of Evolutionary Computation. Genetic Algorithms view a solution as belonging to a species that can evolve over generations. Genetic Algorithms mimic biological evolution to generate better solutions from existing solutions through survival of the fittest and mutation and crossbreeding. A sequence of genes makes up a chromosome. Chromosomes are a possible solution and alleles are the possible values of a gene (Khan).

Each dance step category has a different allele number and there are four genes in each of my chromosomes; they equal a sequence of dance steps.

Related/Inspired work:

The inspiration for my project is Wayne McGregor’s collaboration with Google in dance and Artificial Intelligence. McGregor’s project is called Living Achieve, it is a choreography tool powered by machine learning. The AI capable of predicting dance moves in his particular style (Deepres). Related work can be found by Patrick DeKelly. Patrick DeKelly used a two-hour video of a dancer’s silhouette and an artificial intelligence to learn each image of the video and create a new dance video silhouette using the images. Neuro-linguistic programming techniques (NLP) and image Convolutional neural network (CNNs) techniques are combined to have a computer learn to make its own dance videos (DeKelly). They took the frames from a dance video of silhouettes, preprocessed them to smaller and simpler, and added them to a zip file in sequence. They compressed the frames with an Autoencoder into a much smaller NumPy array. They put these compressed frames into sequences and trained a model to create more (DeKelly.

Methodology:

A person doing a handstand on a track

Description automatically generated with low confidenceA person doing a handstand on a track

Description automatically generated with medium confidenceA person running on a track

Description automatically generated with medium confidence

Upon setting up the suit, I recorded myself doing 63 common dance steps and separating them into their given category. Some dance steps had to be modified due to where the sensors were located on the body. The knee sensors made it difficult to do floorwork or the battery pack made it difficult to do leaps. I saved all the motion capture files as bvh files and connected them to my comma separated file by their location.

These rules were created inside of a function to figure out the layout of a dance sequence:

Text

Description automatically generated with low confidence

These rules were used to optimize fitness of a sequence. For each rule that applies to the genes in a chromosome, the fitness for the chromosome would decrease. If the fitness value equals 1.0 the algorithm breaks, if not the program continues. A new tuple was appended to *Solutions* of 4 dance step types to a list and get 50 possible solutions (The Builder). This was done by importing a random integer between 0 and 5 because these are the categories of dance steps. Now creating the genetic algorithm starts. Elitist selection was used to find the fittest sequences. All 50 solutions were ranked in a *RankedSolutions* list and appended the fitness value and the tuple of dance categories. *RankedSolutions* was then sorted and reversed the order so that the indices with the highest fitness values would be at the top (The Builder). This led to the problem of there not being enough diversity because it caused dance sequences that started with 4 to go to the top because of their rank. The top 10 solutions were taken from *RankedSolutions* and moved to the *BestSolutions* list. This is where the break would happen. A random value you be selected from BestSolutions and if fitness equaled one the genetic algorithm would give that tuple.

Text

Description automatically generated with medium confidence

For each value in *BestSolutions* the values of the tuple were extracted and put in an *Elements* list. Then the *newGeneration* list was created and each new allele was picked randomly from *Elements* and mutating them by multiplying them by a random integer and using modulus 5 so that they stay within the range or 0 and 4.  The new elements were appended to the new generation and make *Solutions* now equal to *newGeneration* (The Builder)

A screenshot of a computer

Description automatically generated with low confidence

After saving the Python file for the Genetic algorithm, a python file was that imported the Genetic algorithm. This python file reads from the comma separated file that contains all the motion capture bvh files, step names, and step types. A for loop was created that separated all steps into their given step type. In this python folder, a method was created that would output a final dance sequence when you told it how many sequences you needed.

Text

Description automatically generated

This was achieved by creating a for loop that iterates through the input number. Each time creating an occurrence of my genetic algorithm getting the list and then appending each index to a new list.  From there another for loop was created that depending on what the number value was of the indices would append a random dance step from the correct category list to the final list.

An Anvil application was created that received user input and connected to the project. With Anvil, a programmer can build the user interface of a web app in Python. After importing and connecting Anvil, a method was made that was reachable by the anvil app. Text

Description automatically generated

Anvil pulled the method, given to get the final dance sequence. If the response the Anvil application receives is not null. It displays it as text in a textbox.

Results:

Graphical user interface, application

Description automatically generated

The results are an Anvil app that will turn the number of sets the user asks for. The only break in the rules given I could possibly see is the connecting of a step four and step one of different sequences. Because the last gene of the chromosome in the genetic algorithm is not checked before reiterating through the code. Other than that, if the user asks for 2 sets of dance sequences it will give them 8 dance steps that within their set of 4 follow the rules given to be a perfect set. Anvil application: <https://dancestepgenerator.anvil.app>

Conclusion:

In conclusion, I was able to create an application. This was done using a genetic algorithm and finding the fitness of sequencing to give a user the best sequence to use in a dance, combo or across the floor passes. I would like to continue looking at the current program with different parameters and setting more rules in the function and make the population bigger. I am also thinking about incorporating the dataset in when creating the more rules to the function inside the Genetic Algorithm. This would allow me to make specific rules for each individual dance step.

Future Work:

To build onto this current project and create future projects with complexity, I would like to create an Azure classification cluster model that would be able to look at the motion capture data that I have and be able to put it in the correct category. For example, if both feet leave the ground for an extended amount of time this dance step would fall under the Leaps and Jumps category. My long-term goal is to create the same system using a long short-term memory system and a recurrent neural network within an encoder-decoder model to create a dance sequence. I would like to continue to work with the motion capture by taking multiple motion-capture files that will generate a new dance; similar to McKelly’s video silhouette’s. I want to push the boundaries to see how creative AI could be when learning given material.

  I would like to thank the SRACE Grant and CSU for allowing me to create a project that combined my Computer Science major and Dance minor studies. I have enjoyed creating a tool to help myself and my peers. The enjoyment received while working on this project has geared me to want to pursue AI further in graduate studies.

Works Cited

The Builder, director. *Genetic Algorithm In Python*. *YouTube*, YouTube, 10 Sept. 2020, [www.youtube.com/watch?v=4XZoVQOt-0I](http://www.youtube.com/watch?v=4XZoVQOt-0I).

Deepres, Ravi. “Living Archive.” *Studio Wayne McGregor*, 16 Jan. 2016, waynemcgregor.com/research/living-archive.

DeKelly, Patrick. “How to Teach an AI to Dance.” *Kaggle*, Kaggle, 12 Dec. 2018, [www.kaggle.com/valkling/how-to-teach-an-ai-to-dance](http://www.kaggle.com/valkling/how-to-teach-an-ai-to-dance).

Khan,Sharmin.Topic\_5\_Genetic\_Algorithm.24 March.2020

Nicholls, Crystal. “This Documentary Will Make You Fall in Love With Jazz Dance All Over Again.” *Dance Magazine*, 16 July 2020, www.dancemagazine.com/jazz-dance-2646414724.html?rebelltitem=1#rebelltitem1.