

Composición Musical a Través del Uso de Algoritmos Genéticos

Ezequiel Moldaver¹, Hernán Merlino^{1,2}, Enrique Fernández^{1,2}

1. Cátedra de Sistemas de Programación no Convencional de Robots. Facultad de Ingeniería. Universidad de Buenos Aires

2. Laboratorio de Investigación y Desarrollo en Arquitecturas Complejas (LIDAC). Grupo de Investigación en Sistemas de Información (GISI). Universidad Nacional de Lanús. Argentina
hmerlino@gmail.co

Resumen—Este trabajo se enfocará en el uso de los algoritmos genéticos (AAGG) con el fin de mezclar armonías y melodías de forma que se genere una composición musical de buen sonido para el oído, lo que significa que el contexto de cada nota respaldará la sonoridad de la misma provocando que no se genere un efecto disonante de forma permanente, que se genere una disonancia momentánea es permisible ya que es parte de la misma música generar tensión a través de pequeños intervalos poco agradables al oído.

Palabras Clave—algoritmos genéticos, música, redes neuronales, composición.

I. INTRODUCCIÓN

La composición musical es el arte que tiene como objetivo la creación de obras musicales, es una actividad humana que sirve para la expresión, comunicación y entendimiento entre personas. Se podría dividir en 2 partes básicas: la melodía y la armonía. La primera, según la Real Academia Española (RAE)[1] es “La parte de la música que trata del tiempo con relación al canto, y de la elección y número de sonos con que han de formarse en cada género de composición los períodos musicales, ya sobre un tono dado, ya modulando para que el canto agrade al oído”, siendo la segunda “la unión de tres o más sonidos simultáneos, entendiendo que el canto o melodía producido por una sola voz es homónimo, con dos voces se producen intervalos armónicos y, a partir de 3 voces o sonidos simultáneos hablamos de armonía”.

Si bien los AAGG [2-3] no fueron diseñados para imitar comportamiento humano sino para resolver problemas u optimizar soluciones, se puede pensar en la composición de la música como el problema de encontrar una canción que suene bien al tomar en consideración el conjunto de todas las composiciones posibles como el espacio de soluciones, este espacio no está estructurado, es decir, que las buenas soluciones pueden estar junto a las malas; al cambiar algunas notas clave en una pieza, ésta puede llegar a ser menos interesante, aunque parezcan parecidas. La idea principal es mezclar de forma aleatoria distintos temas musicales de forma iterativa, en cada iteración se seleccionarán las mejores canciones para ser la base de la iteración siguiente. Además, en base a una variable probabilística se realizará un cambio aleatorio sobre uno de los compases de alguno de los temas, que también será elegido al azar. Luego de varias repeticiones se obtendrán como resultado distintas piezas musicales que serán totalmente distintas a las primeras, generando de esta forma, nueva música.

Con el fin de priorizar la estructura de temas agradables o buscados por el usuario se utilizarán o un sistema de premios y castigos o redes neuronales (RRNN) [4-5] que serán entrenadas

para permitir un procesamiento automático de composición basado en sus gustos e inclinaciones musicales. Las mismas serán entrenadas con este fin para interpretar de todas las canciones candidatas en cada iteración y seleccionar aquellas que, de acuerdo a la estructura generada por el entrenamiento, sean las que el usuario hubiese elegido.

II. OBJETIVOS

El proyecto tiene diversos objetivos que se encuentran orientados a las distintas aristas que tiene la composición musical. Cuando un músico o compositor musical se encuentra sin inspiración permanente puede utilizar como método de ayuda para superar esta situación el compositor musical a través de AAGG. El primer objetivo de todos los que se van a plantear es que las canciones generadas por la herramienta sean agradables al oído, es decir, las partes disonantes deben ser armónicas y deben ser una minoría en proporción a las que no. No tendría sentido que una persona componga un tema que sea desagradable al escucharlo, si bien hay distintos géneros musicales y existen personas que no les gustan unos y si otros, dentro de cada género existen ciertos comportamientos intrínsecos a él que hacen que éstos tengan una estructura más o menos definida y, dentro de esta, serán agradables al oído de las personas a las que les gusta ese tipo de música. El segundo objetivo es netamente legislativo, para proteger la propiedad intelectual de los compositores. Las canciones generadas por el algoritmo de composición musical no deben ser plagio de las partituras que les dieron origen. Se fijará un criterio el cuál debe cumplirse estrictamente para no ser victimarios del delito de plagio, respetando de esta manera a toda la comunidad artística de la cuál se forma parte. Se debe ser muy cuidadoso a la hora de producir un nuevo contenido artístico ya que este puede incurrir en el delito de violación a la ley 11.723 de Propiedad Intelectual de la República Argentina, por estos motivos se fija este objetivo completamente orientado al proteccionismo y al respeto. Como tercer parte este programa debe respetar los gustos musicales del usuario, se buscará que la respuesta del sistema sea una clara consecuencia de la entrada que se le de al mismo, suponiendo de esta forma que el usuario quiera usar como base generadora de su composición las partituras iniciales, manteniendo ciertas características de los mismos, viéndose afectadas en parte, lo suficiente como para generar una canción que cumpla con el segundo objetivo pero a la vez se cumpla con el que se esta definiendo en este punto. Por último, el software debe generar una partitura con ciertos criterios definibles por el compositor, todas las transformaciones que sufran las partituras de entrada al sistema deben estar reguladas a gusto y capricho del usuario, pudiendo definir distintos tipos de restricciones o reglas a través de las

cuáles se premien o se castiguen ciertas mezclas o características musicales que se puedan ir dando a lo largo del proceso iterativo de composición, cumpliendo así con las metas que esté buscando el usuario en función de poder manejar algunas de las variables que se encuentran libradas al azar dentro de la ejecución del programa.

III. PROBLEMA A RESOLVER

Un músico muchas veces cae en mesetas de inspiración que limitan su producción musical, la mayoría de las veces esto sucede luego de haber tenido una etapa muy fructífera en cuanto a composición. Además, un compositor sufre de momentos efímeros de iluminación que muchas veces terminan decantando en pequeñas piezas musicales que quedan condenadas, en general, al olvido.

Por ello, se necesitaría implementar una solución que permita obtener un material en crudo para ser escuchado, estudiado, analizado y, finalmente, modificado a criterio del músico con el fin de servirle a éste de ayuda para los momentos de poca inspiración. Este material debe poder obtenerse de manera flexible tanto de las pequeñas piezas que se componen como de canciones completas ya que no siempre se cuentan con las primeras.

Dado que en la música oriental existen 12 notas musicales y distintos tipos de combinación entre ellas es difícil para un algoritmo saber cuando una pieza suena bien o no, además existen estudios llamados de armonía que se encargan de este tipo de cuestión. Si bien existen las disonancias, sonidos tensos o poco agradables al oído, éstas enriquecen la composición quitándole la monotonía de seguir algún tipo de escala conocida para ese tipo de estructura de canción. Además, las mismas pueden ser una introducción a un cambio de tonalidad haciendo de “pase” o puente entre una y otra. Por último, queda señalar, que siempre la música queda reducida al criterio del compositor, lo que busca, siente y quiere expresar, no existen reglas que descarten la disonancia, pero por otro lado a nadie le gustaría una canción que en cada una de sus partes haya tensión o sea totalmente disonante.

Por una cuestión de respeto tanto a la comunidad artística como al público y de idoneidad, una canción no debe ser un plagio de otra. No existe un criterio uniforme que determine cuando una pieza musical ha copiado a otra, esto genera varias preguntas:

1. Con que tenga alguna mínima parte igual ya implica que un artista a copiado a otro?
2. Qué significa una mínima parte? Cuánto es?
3. Cómo se determina esto en los géneros más populares dónde las canciones son más parecidas entre ellas?
4. Qué y cuánto tienen que tener de distinto para diferenciarse?

Se podrían escribir muchos interrogantes similares que nos den cuenta que este es un problema complejo y difícil de solucionar. Tanto es así que en los juicios sobre plagio no se aplica otro criterio que no sea el de los peritos musicales, no hay un método o regla exacta que nos permita descubrir o determinar si se ha incurrido en una violación a la propiedad intelectual.

Cada artista tiene su estilo bien marcado, esta esta característica no puede ser simplificada a algunas pocas formas y recursos musicales, es mucho más complejo que eso. Se sabe que la búsqueda de una identidad musical le lleva varios años de trabajo y esfuerzo a cualquier compositor o músico de cualquier género. Además, una vez que se encuentra el mismo,

no se deben estancar en la simplicidad de haber llegado a su (primera) meta, el camino recién empieza, ahora quedará la profundización del concepto, las variaciones para no caer en la monotonía, la evolución de lo que se ha encontrado, etc. es un trabajo arduo y continuo que no tiene un fin determinado.

Más allá del género o estilo musical que se este buscando respetar, generar o lograr, el compositor puede estar interesado en que no estén presentes ciertas características musicales en función de sus objetivos. Que el usuario pueda tratar de evitar una conjunción de notas disonantes, exceso de silencio en los compases, muchas idas y vueltas en cuanto a los tiempos musicales a lo largo de la canción y la cantidad de notas máximas y mínimas por compás es fundamental para poder llegar a las metas que un músico se propone cuando busca algún tipo de inspiración a partir de la cuál desarrollar sus habilidades. Este problema también se puede plantear de la forma inversa, es decir, que en vez de que se evite que se formen ciertas particularidades en las partituras generadas, se desee fomentarlas.

IV. SOLUCIONES A LOS PROBLEMAS PLANTEADOS

Con el fin de solucionar el primer problema planteado, poder generar temas musicales agradables al oído, se ha pensado en que la mejor solución para no caer en la rigidez y la monotonía es usar el criterio del usuario. Esto no significa que el mismo tendrá la ardua tarea de identificar en cada paso de cada iteración si las canciones que se están generando son o no agradables al oído, por el contrario, esto implicará que el compositor defina cuáles son las primeras soluciones al problema de composición para el algoritmo genético y, de esta manera, ya estará dando por sentado las bases de las posibles combinaciones entre partituras que se den en las distintas instancias de la evolución en el proceso de composición automática. Si bien no se puede planear o imaginar todas las combinaciones posibles que pueden tener las piezas musicales entre ellas, se puede hacer un análisis general sobre el problema a través de los tonos de las canciones. Hay tonos que pueden combinarse con otros sin ser disonantes, otros que generan una disonancia de tensión que puede enriquecer mucho el tema y otros que finalmente no podrían mezclarse porque no podría cumplir con la regla de generar canciones agradables al oído. La cantidad de temas, su longitud, sus variaciones, sus tonos, todas sus características y sus proporciones quedarían en manos del usuario en función de establecer y sentar las bases para la composición.

En función de no caer una violación a la propiedad intelectual el algoritmo genético no sólo irá mezclando los distintos compases de las diferentes canciones, sino que también, aplicará una mutación o modificación de los compases de manera estadística que producirá una gran variación de las canciones de las cuales partimos. De esta forma y, a partir de las iteraciones evolutivas que se establezcan, se irá modificando paso a paso cada una de las piezas musicales que se encuentran como parte del proceso de composición automática. Dado que estas variaciones no son determinísticas en cuanto a cuándo se producen, sino que por el contrario es un proceso estocástico, otra vez se verá involucrado el usuario para definir una probabilidad de ocurrencia de este fenómeno.

En función de poder respetar los gustos o preferencias musicales de un usuario se definirá la implementación de un sistema de premios y castigos para poder fomentar y desalentar ciertas características o propiedades de acuerdo a lo que el usuario esté buscando. Si se desea componer un tema de rock

se alentará que estén presentes todas las particularidades que posee el género, en cambio, si se busca formar una pieza musical de jazz, se pedirá que las combinaciones que se premien sean otras totalmente distintas. Por este motivo, se intentará generar una cantidad de reglas genéricas que abarquen un gran espectro de posibilidades y, que de acuerdo a su combinación y especificación le permitan al compositor poder expresarse con mínimas limitaciones para poder llegar a su objetivo. Si bien el sistema de restricciones definido por las reglas que exprese el usuario no es determinístico en función de lo buscado, si lo será en función de las canciones que mejor se adaptan a éstas.

El compositor puede estar buscando que su composición se encuentre sesgada por ciertos rasgos que no son comunes a un género o estilo musical, para ello sería ideal poder contar con una herramienta que con tan solo recibir partituras que posean estos rasgos extraiga su idea principal, musicalmente hablando, con el fin de plasmarla en la composición final. Como un ejemplo de ello se podría decir que se está queriendo llegar a componer una canción que sea un HIT, esto no es propio de ninguna rama de la música. Para poder lograr la solución a este problema se puede entrenar una red neuronal que procese el tipo de canción que el usuario especifique y obtenga un modelo con tal de respetar el criterio definido musicalmente. Como contraste a esto, también se solicitará al compositor que indique las partituras que están alejadas de los objetivos o no cumplan con la idea musical que se desea encontrar.

V. TRABAJOS REALIZADOS EN ESTE CAMPO

Se han desarrollado varios estudios en base al estudio del arte computacional [6-11], en particular para este caso aplicaciones de desarrollo de las soluciones RRNN y AAGG para la composición musical [12-34] pero todos de diferentes enfoques, análisis y conclusiones; incluso hasta se han grabado discos y se ha promocionado este tipo de arte tecnológico. Los enfoques más importantes son:

- Gibson and Byrne [35] hicieron un trabajo basado en la armonización de melodías usando solamente las notas tónicas subdominantes (la cuarta nota de una escala musical, Ej. en la escala de Do la subdominante es el Fa) y dominantes (la quinta nota de una escala musical, Ej. en la escala de Do la dominante es el Sol) de los acordes pertenecientes de las melodías estudiadas.
- Jacob [36] presentó su trabajo de composición en el cuál los AAGG son usados para identificar y diferenciar melodías aceptables para el oído humano, las mismas son tomadas de la salida que produce un proceso de generación de música estocástico. A partir de esa elección comienza el proceso de mejora y refinamiento propio del AG. McIntyre [37] realizó estudios de armonización de 4 partes usando una armonía barroca.
- Uno de los proyectos más conocidos es GenJam (Biles) [37-41] un AG que genera solos de jazz en base a una sucesión de acordes. Establece escalas para la improvisación de acuerdo al acorde utilizado en el momento. Dada la progresión de los mismos se analiza y se establecen las notas en base a las escalas musicales. Mientras la aplicación da una muestra en tiempo real de como quedaría el solo, el usuario aprueba o rechaza la misma dando de esta manera un entrenamiento a una red neuronal que luego tomará las decisiones por el usuario de forma automática.
- Se ha hecho un trabajo similar (George Papadopoulos and Geraint Wiggins) [42] para establecer una melodía dada la

progresión de acordes que se recibe, en este caso se establece una función de aptitud y no una red neuronal para elegir a los individuos más aptos. Esta investigación se focalizó en intentar reproducir el comportamiento humano de la composición por lo que se utilizan una función de aptitud basada en conocimiento psicológico musical.

Este trabajo se diferenciará de los dos últimos mencionados ya que se enfocará en la composición musical desde todos los puntos de vista y maneras de trabajar, no sólo se podrán componer nuevas melodías sino que también nuevas progresiones de acordes. También se trabajará con distintos tipos de métodos en la determinación de los individuos más aptos, lo que llevará a un entendimiento con el usuario para adaptar los AAGG a su gusto. Finalmente se evaluará a la partitura generada determinando si ésta es realmente una nueva pieza musical de forma genuina y auténtica o es un plagio de alguna de las partituras originales.

Una característica a destacar es que en la forma en la que trabajaremos podremos utilizar como fuentes de composición melodías o progresiones de acordes que pertenezcan a cualquier ritmo musical lo que aporta una riqueza en el producto final que no se posee en los trabajos hasta ahora mencionados.

VI. DESARROLLO DE LA SOLUCIÓN

El objetivo de este trabajo es romper totalmente con los esquemas de investigación y desarrollo que se han utilizado en el área, intentando componer, tal como lo hace una persona, nuevas piezas musicales. Para ello se trabajará con una cantidad inicial de partituras de temas ya existentes o recién compuestos que en los que se quiera investigar, sobre ellos se considerará a cada compás como un cromosoma del individuo interviniente y entre ellos se realizará la cruce. Luego de varias iteraciones cada uno de los temas originales se verán totalmente modificados en estructura, tiempos, ritmos, sonoridades, etc. generando de esta manera canciones nuevas. De la morfología, estructura y sonoridad de éstas puede nacer la inspiración que busca el compositor, o, directamente una nueva composición musical de manera automática. Además, los AAGG, al depender del azar pueden producir muchas soluciones diferentes, lo que es necesario en los ambientes creativos.

Como función de aptitud (para elegir las mejores canciones de cada iteración) se utilizarán tres posibilidades distintas:

- Un archivo de configuración en el cuál se especificarán ciertos parámetros asociados a las combinaciones musicales buscadas pensando las que no se produzcan en ese sentido.
- El usuario podrá entrenar una red neuronal que permita discernir que melodías son las buscadas y seguirán en pie dando origen a la población siguiente y que melodías pasarán al olvido.

Se usó un software open source para utilizar su modelado, abstracción, gráfica, etc. para la escritura de las partituras musicales a procesar. En el mismo también se diseñan, escriben y guardan los archivos de datos correspondiente a la entrada de la aplicación a desarrollar. El archivo de salida será de igual formato que los que ingresan, para ser reproducidos, nuevamente se apelará al editor de partituras en cuestión. Por estar desarrollado en el lenguaje de programación Java, ser un software OpenSource, su simpleza en cuanto a la usabilidad, su capacidad de abrir y guardar en los formatos más conocidos de partituras, se eligió el TuxGuitar como editor de partituras.

Se tomará el siguiente criterio de plagio: si las piezas resultantes contienen hasta 8 compases idénticos, sin importar el orden, a alguna de las melodías que les dieron origen, se considerará que no a compuesto música, que se está plagiando; en caso contrario diremos que el resultado es genuinamente original.

Se podrán mezclar cualquier instrumento excepto la batería ya que no sigue las mismas reglas de escritura musical de una partitura que cualquiera de los otros instrumentos que existen. El formato para escuchar las canciones resultantes será mediante el protocolo de comunicación MIDI.

Se desarrolló un sistema que en base a una entrada de partituras musicales genera, de acuerdo a la función de aptitud elegida, una partitura sin violar la ley de propiedad intelectual con respecto a las originales, que es la que mejor se adapta según lo seleccionado por la función de aptitud.

El primer caso es un sistema de penalizaciones que se aplica cuando se dan ciertas condiciones. Todas las canciones empiezan con un puntaje determinado y se le van restando las penalizaciones, el individuo más apto será el que tenga mayor cantidad de puntos. Vale aclarar que para hacer el camino inverso al de la penalización, es decir, la premiación de alguna condición o característica musical presente, se puede penalizar con un valor menor a cero, por lo que la misma se transformaría en una adición de puntos en lugar de una quita. Esta funcionalidad inversa puede ser aplicada a cualquier tipo restricción.

Las 3 condiciones que hoy se contemplan son las siguientes:

1. Restricción de notas: penaliza que en un mismo compás se encuentren dos notas especificadas (expresadas en notación americana, DO-RE-MI-FA-SOL-LA-SI es equivalente a C-D-E-F-G-A-B), se evalúan para todos los compases de la partitura. Se define la penalidad y las dos notas involucradas.
2. Restricción de tiempos: penaliza que en una misma partitura se encuentren los dos tiempos especificados (Ejemplo 2/4, 4/4, etc). Se establece la penalidad y ambos tiempos.
3. Restricción de cantidad de notas: penaliza la cantidad de notas, ya sea mínima o máxima, que se encuentran en un compás a lo largo de toda la partitura. Se indica la penalidad y la cantidad de notas.

El segundo tipo de función de aptitud se establece a través de una red neuronal, puede ser una que ya hallamos entrenado o se puede entrenar una nueva red. La misma es del tipo Backpropagation y tiene 4 capas, una de entrada, dos ocultas y una de salida. La primera capa tiene la cantidad de neuronas igual a la menor cantidad de compases de las canciones con las cuales se entrenó, la segunda y la tercera capa son configurables, la última sólo tiene una neurona que indica de 0 a 1 la preferencia de ese tema para continuar en la siguiente iteración del algoritmo genético.

Para que la red pueda ser entrenada hay que pasar la dirección a una carpeta que contenga las partituras que representen los objetivos buscados y otra a una carpeta que contenga las canciones que poseen propiedades que no nos interesan de un tema. Luego de ser entrenada la red se guarda en disco para que pueda ser utilizada más adelante.

A. Armado de la Red Neuronal

Para serializar los datos hay que tener en cuenta que en una partitura todos los datos que se pueden leer se pueden asociar y

combinar sin ninguna limitación, desde las notas propiamente dichas (es “infinito” cuán aguda o grave es una nota, la única limitación aquí es el instrumento, cuántas teclas tiene el piano, cuántos trastes la guitarra, el largo del violín o el chelo, etc), la cantidad de notas (en un compás o que suenan en simultáneo), sus tiempos asociados (redonda, blanca, negra, semi corchea, etc) hasta la cantidad de compases. Dadas las características de estas variables se diseñó una manera de serializar los datos que tenga en cuenta todos estos aspectos.

Se tuvieron que plantear ciertas limitaciones para establecer el universo en el cuál la aplicación se va a manejar. Las mismas son:

- Se determinó que habrá una relación unívoca entre neuronas de entrada y compases de las canciones, al ser un número variable entre todas ellas se fijó que se evaluarán solamente los compases de la canción que tenga menor cantidad de ellos, todos los compases que superen ese número no serán evaluados.

Ejemplo: Si tenemos 3 canciones: A, B y C que tienen respectivamente 50, 60 y 100 compases solamente se tomarán en cuenta los primeros 50 generando una red neuronal con ese número de neuronas en la primera capa.

- Se fijó la nota más grave y la nota más aguda de acuerdo a un instrumento de referencia, en este caso, la guitarra. Por lo que por su conformación la nota más grave es un Mi mayor tocando la sexta cuerda (la más gruesa) sin presionar ningún traste, lo que comúnmente se llama “al aire”. De la misma forma, por una limitación física del instrumento, se establece la nota más aguda que se puede producir, que en este caso, se logra tocando la 1er cuerda (la más fina) en el último traste, comúnmente, el número 29.
- Por una cuestión de simplicidad a la hora de abordar el problema se decidió ignorar en la serialización los efectos que una nota o un conjunto de notas pueda llegar a tener.
- Dado que una partitura puede tener escrita todos los instrumentos y no es compatible mezclarlos deliberadamente se toma en cuenta solamente el primer track de la canción, es decir, que todos los tracks más allá del elegido para ser evaluado serán ignorados.

A partir de estas bases se estableció un número correspondiente a cada nota, para la más grave un 1, para la siguiente un 2 y así sucesivamente hasta alcanzar el número máximo alcanzable por el instrumento. En la práctica se observó que la red funcionaba mucho mejor con datos normalizados por lo que finalmente la nota serializada es: $N^{\circ} \text{ nota} / N^{\circ} \text{ nota máxima}$. En cada compás se suman todas las serializaciones de las notas generando la entrada de datos para la red. Cuando la nota sea un silencio se tomará el valor cero para el mismo.

Usando esta configuración y forma de manejar los datos se tienen en cuenta la cantidad de notas que se encuentran en un compás y sus valores intrínsecos. También se valora, de acuerdo a la escala normal utilizada, las octavas de preferencia de las notas; no es lo mismo un Mi grave que uno agudo, esto se produce al tener una distinta representación numérica como dato de entrada serializado para el uso de la red neuronal. Los valores más agudos tendrán un valor más cercano a uno dado que se producen a medida que se avanza sobre el traste de la guitarra y los sonidos más graves serán representados por números cercanos al 0. De forma menos ortodoxa también es tenido en cuenta el tiempo de cada nota y el del compás en la modelización en cuestión, esto sucede porque musicalmente cuando se define un tiempo para un compás o serie de

compases se limita el tamaño del tiempo de las notas, si el tiempo de un compás es definido en 4/4 entran o 4 negras o 2 blancas o 1 redonda y si se lo define en 3/3 se pueden escribir o 3 negras o 1 blanca y una negra pero no una redonda; entonces al sumar cada nota del compás habrá distintos valores medios por compás de acuerdo a este tipo de característica.

La elección de la función de transferencia fue un paso importante para este proyecto, de acuerdo con las variantes que había disponibles se eligieron las que podían resolver mejor este problema y, empíricamente se hicieron distintas pruebas para determinar la mejor opción de todas. Las posibilidades más importantes eran:

- Lineal $y = b * x$: Es la función de transferencia más simple que existe, se transfiere a la siguiente capa el valor de entrada aplicándole solamente una transformación lineal. Este tipo de función se utiliza generalmente para transferir una entrada de datos a dos capas paralelas de procesamiento, este uso se muestra en el siguiente diagrama (figura 1):

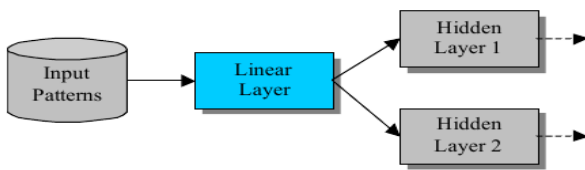


Fig. 1. Diagrama de conexión del Linear Layer.

- Biased $y = x + \text{bias}$ (bias es el bias de la enésima neurona): Esta tipo de transferencia es una extensión de la lineal porque aplica una transformación lineal pero a la vez difiere en dos aspectos:

1. Al usar biases la transformación se va modificando durante la fase de entrenamiento.
2. No tiene un multiplicador como parte de la función.

- Sigmoid $y = 1/(1+e^{-x})$: Aplica una transferencia no lineal con un rango de datos que va desde cero a uno, es una buena función para ser utilizada en las capas ocultas de una red neuronal.

- TanH $y = (e^x - e^{-x}) / (e^x + e^{-x})$: Es similar a la Sigmoid, sólo que se aplica la tangente hiperbólica y por lo tanto la salida quedará limitada en un rango comprendido entre menos uno y uno.

- SoftMax $y = e^x / \sum_{j=1}^c e^x$: Como la Sigmoid tiene una salida que cubre el rango entre cero y uno, pero la gran diferencia es que la suma de todas las neuronas de la misma capa suman 1. Gracias a esta característica la salida de esta función de transferencia puede ser interpretada como probabilística. Su uso más frecuente es en redes neuronales supervisadas para la clasificación entre C elementos.

$$y = \log(1+x) \text{ si } x \geq 1$$

- Logarítmica $y = \log(1-x) \text{ si } x < 1$: En este caso el resultado de la función va de cero a infinito. Esta relación nos permite evitar la saturación de procesar muchos elementos entre cero y uno, donde la Sigmoid y la TanH tienen una curva de respuesta de muy poca pendiente.

- Seno $y = \sin(x)$: Este tipo de transferencia suele utilizarse para problemas que poseen cierta periodicidad en su estructura. El rango de salida es de menos a uno.

La función de transferencia que se utilizó para resolver el aprendizaje sobre la composición musical buscada fue la Sigmoid.

La estructura interna de la red depende de la cantidad de información que debe aprender, por lo que para hacerla más flexible se pueden configurar el número de neuronas de las 2 capas ocultas de la misma. Empíricamente se comprobó que para “memorizar” los 40 primeros compases de entre 25 y 30 canciones se necesitan 10 neuronas en cada una de las capas.

La conformación de la cantidad de neuronas por capa de la red siempre sigue el siguiente patrón:

1. Capa de entrada de datos: La mínima cantidad de compases que posean de los temas de aprendizaje.
2. Primera capa oculta.
3. Segunda capa oculta.
4. Capa de salida de datos: La última capa se resume a la probabilidad de que ese tema sea el buscado. Si el número es cercano a uno estamos en presencia de patrones musicales de agrado para el usuario, para el caso contrario el límite es el cero.

La morfología de la red neuronal depende del entrenamiento que se le dé a la misma (dado que en ese momento se establecerá la cantidad de neuronas de entrada, es decir, la cantidad de compases que se tomarán en cuenta para la evaluación), la configuración que se especifique para las capas ocultas y, finalmente, la capa de salida de datos que revelará la probabilidad de que ese tema contenga la idea musical que se está buscando. Esta flexibilidad que se posee permite al usuario afrontar problemas de menor y mayor embergadura en cuanto se refiere a la composición musical buscada solamente cambiando el archivo de configuración que recibe la aplicación. Si se desea trabajar con estilos de música cuyos temas en general son de duración extendida sólo se debe agregar más neuronas en las capas ocultas ya que para la capa de entrada se identifica automáticamente la cantidad de neuronas necesarias para llevar a cabo su procesamiento; en cambio, si se desea trabajar con un universo de temas musicales cuya característica es tener una corta duración sólo se debe disminuir el número de neuronas de las capas ocultas.

VII. VALIDACIÓN DE LA SOLUCIÓN

Se han diseñado 2 casos totalmente distintos para probar la solución desarrollada. La idea básica es probar las distintas aristas a nivel de composición que brinda la herramienta, validando tanto el método de penalizaciones como el de aprendizaje vía redes neuronales.

En todos los casos se evaluará la existencia o no de violaciones a la ley de propiedad intelectual, ya que el propio sistema en la salida indica si se ha caído en el plagio.

Se cuenta con una amplia base de datos de partituras de canciones que se deben adaptar a lo que se está buscando. Dado que siempre se mezcla la primera pista de partituras se debe proceder a editar la misma y colocar la pista con la que se desea trabajar como track principal. Una vez hecho esto se puede hacer un pulido más fino eliminando algunos silencios y/o corrigiendo algunos compases en función de las tonalidades de las demás partituras o en base a lo que se está buscando.

Cabe aclarar que para utilizar el software y tener respuestas satisfactorias, se deben tener muy en cuenta las soluciones iniciales que se le brindan al algoritmo genético, ya que a partir de ellas se delimita el universo de las posibles soluciones, estableciendo de esta forma un espectro de resultados posibles.

A. A través de penalizaciones

Usando el sistema de penalizaciones como función de aptitud se hicieron 50 pruebas distribuidas en 10 casos distintos con 5 repeticiones por caso. La definición de los 10 casos se dieron en base a los siguientes parámetros:

- Probabilidad de mutaciones.
- Cantidad de evoluciones.
- Factor de cruza.
- Partituras iniciales.
- Restricciones de cantidad de notas (tanto máximas como mínimas).
- Restricciones de tiempos.
- Restricciones de mezcla de notas específicas por compás.

Los distintos parámetros se eligieron en función de las características musicales de las partituras con las que se trabajó. Pues no tiene sentido alguno establecer restricciones de tiempos que no existan en las partituras o penalizar cierta conjunción de notas dentro de un compás cuando dicha combinación notas tampoco podría producirse.

Lo primero que se puede destacar de las validaciones realizadas es que es raro que las soluciones de composición musical automática se repitan con frecuencia, sólo ocurrió el 4% de los casos, en otras palabras, de las 50 canciones generadas únicamente hay 4 que son iguales, lo que representa una repetición en 2 casos.

Lo segundo para destacares que más allá de las restricciones que se pongan en cuanto a la cantidad de notas máximas de un compás un factor importante en este aspecto es la probabilidad de mutaciones con respecto a la cantidad de evoluciones, ya que esta mezcla 2 compases al azar y genera que se sumen la cantidad de notas que poseen ambos. Entonces, al haber una gran cantidad de iteraciones sobre el mismo espacio de canciones genera que inevitablemente se den estas condiciones a pesar de su penalización. Esto se debe a que la penalización pierde su poder restrictivo en el sentido mas estricto porque todas las canciones habrán tenido muchas notas en un mismo compás.

Otro factor que merece ser mencionado es que si se superan las 200 evoluciones la probabilidad de plagio se reduce muchísimo y se coloca en el 2% mientras que por debajo de este nivel es críticamente alto, llegando al 40% de plagio sobre las partituras originales. Esto se debe a que al haber más iteraciones sobre el algoritmo genético se generan más cruza entre canciones y más mutaciones, por lo que se van mezclando hasta perder su morfología original.

El último tema a discutir es la sonoridad generada en las soluciones que se obtienen. En este sentido se observar que la misma depende básicamente de la compatibilidad musical entre las partituras originales es un arma de doble filo, porque en un margen mínimo se obtienen resultados interesantes si la disonancia de los tonos a mezclar es mínima, pero si cada tema tiene un tono distinto no quedará nada que se pueda rescatar musicalmente como tal, si podría ser el disparador para el compositor, pero difícilmente salga algo agradable al oído humano. Existen otros factores que afectan en menor medida pero le dan forma al resultado final, ellos son todas las condiciones y características de configuración.

Empíricamente se pudo comprobar que los resultados producidos no pueden llegar a ser en su totalidad una nueva canción, pero si pueden tomarse distintas fracciones de la partitura, realizar conjunciones, variaciones sobre la misma, etc. como disparadores para ayudar al compositor en la ardua

tarea de generar música cuando cae en una meseta de inspiración propia.

B. A través del aprendizaje de gustos musicales

Para generar partituras a través del uso de RRNN se hicieron 10 pruebas. Se tomaron 5 voluntarios, dos de ellos con conocimientos musicales en cuanto a composición, partituras, armonía, etc. y se les propuso hacer 2 pruebas a cada uno. Hay que tener en cuenta que este tipo de pruebas es muy costosa en cuestión de tiempo ya que hay que entrenar una red neuronal para que aprenda los criterios de cada voluntario, luego hay que definir los parámetros del algoritmo y finalmente esperar que se produzcan los resultados; en total, el proceso completo lleva aproximadamente 5 horas ya que el entrenamiento y ejecución de los conocimientos es una tarea muy costosa en cuanto al procesamiento.

Si bien los resultados producidos no son muchos, no se han visto casos de plagio con respecto a las partituras originales. Esto puede deberse ya a que diferencia del caso anterior la función de aptitud es general, es decir, se toma a la canción como un todo para evaluar los gustos de la persona que la entrenó y no restricciones estrictas sobre las distintas características de cada uno de los compases, lo cual genera más flexibilidad ante los cambios que pueda llegar a producir el algoritmo sobre las piezas musicales.

En cuanto a los resultados producidos se puede observar que al igual que en el caso anterior no se podría tomar la pieza musical resultante como producto terminado ya que hay algunas partes de la partitura que no son agradables al oído, pero la mayoría de la partitura corresponde con una composición musical agradable y armónica al oído humano. Sin embargo, los voluntarios aprobaron las canciones que “compusieron” justificando que respetaba en un amplio margen sus gustos y la forma de la cuál habían entrenado la red. Cabe destacar que las soluciones iniciales brindadas al algoritmo genético debían compartir algunas características de las canciones con las que se entrenaron la red ya que el espectro musical es extremadamente amplio y no tiene sentido, por ejemplo, entrenar una red con temas característicos de pop y aplicarla a temas de heavy metal. Esta restricción es en base a cada usuario y se aplica su razonabilidad a la hora de utilizar la herramienta en cuestión. Si, en cambio, se permite cierto tipo de diversidad musical ya que esto enriquecerá el producto final, pero debe ser aplicado con valores razonables.

VIII. CONCLUSIONES

A la primera conclusión que se puede llegar a partir de las pruebas que sean realizado es que el sistema sirve, como mínimo, de manera de disparador de composición para el músico que se encuentra en una depresión productiva, que no se encuentra inspirado. Al utilizar esta herramienta y jugar con ella puede utilizar los compases más jugosos de acuerdo a su búsqueda como primer paso de una canción. Quedará a su criterio completar los baches de poca sonoridad que se pueden generar o seguir ejecutando la herramienta y realizar una conjunción de todos los resultados producidos y así llegar a una composición completamente automática en varios pasos.

Un resultado importante que nos arrojan los análisis realizados es que es difícil que se caiga en una violación a la ley 11.723 de Propiedad Intelectual de la República Argentina, lo cuál es un gran logro ya que sin esto el fin de la herramienta se vería reducido ampliamente. No tiene sentido usar una ayuda para componer que plagie las partituras de entrada del sistema, generaría mucho más trabajo para el músico ya que

debería ocuparse de cambiar la canción generada lo suficiente como para no incurrir en este delito.

Un aspecto novedoso de esta herramienta es que permite convertir a cualquier persona en un compositor musical ya que no es necesario que se tenga conocimientos de música, solamente entender básicamente cómo funciona un algoritmo genético, que ya de por sí es una idea básica que al estar basada en la Teoría de la Evolución la mayoría de las personas conocen. Con indicarle a una red neuronal cuáles son las canciones que a uno le gustan, las que no le gustan y seleccionar una cantidad de canciones para que sean la entrada del programa alcanza para que la implementación de los algoritmos haga el resto y produzca un resultado acorde. De esta manera la herramienta permite universalizar el proceso de composición y ponerlo al alcance de la mano de cualquier persona. Cabe destacar que la música que se genere de este modo respetará lo más posible los resultados musicales buscados por el usuario, dependiendo de lo posible que eso sea de acuerdo al estímulo de entrada que se le brinde. Por lo que esta herramienta no sólo permite convertir a cualquier persona en compositor sino que también reflejará la idea musical del mismo.

Una arista fundamental que posee el sistema de composición es, que más allá de los criterios que el usuario intente tratar de imponer tanto con una red neuronal como con penalizaciones, los resultados siempre estarán sujetos a las partituras de entrada que se le hayan brindado. No tiene sentido tratar de imponer un castigo a la cantidad de notas mínimas o máximas cuando en las canciones que se barajan se violan ambas restricciones, esto provocará que todas las partituras sean penalizadas de igual forma, por lo que no habrá diferencias entre ellas a la hora de seleccionar las más aptas para la siguiente evolución. El mismo punto de vista puede ser aplicado para las redes neuronales, ya que la red intentará memorizar ciertas estructuras, escalas, notas, etc. y si la entrada al programa no tiene nada que ver con el entrenamiento el comportamiento será completamente azaroso. La buena elección de criterios y canciones de entrenamiento con respecto a la música con la cuál se va a trabajar será fundamental para acercarse más y más a las ideas musicales a las que se desea llegar.

AGRADECIMIENTOS

Quiero agradecer a al Mg. Ing. Hernán Merlino y al Dr. Enrique Fernández que sin ellos no hubiera podido desarrollar este trabajo. También me gustaría agradecer a mi familia por el apoyo y a la Facultad de Ingeniería de la Universidad de Buenos Aires que fue mi segundo hogar y me dio todas las herramientas que hoy tengo profesionalmente.

REFERENCIAS

[1] URL: <http://lema.rae.es/drae/?val=melod%C3%ADa> Página válida al 20/09/2011.

[2] Mitchell, M. (1996) An Introduction to Genetic Algorithms. Cambridge, MA: MIT Press/Bradford Books.

[3] Ross, P. M. and D. Corne (1995). Applications of Genetic Algorithms. AISB Quarterly 89, 23-30.

[4] Braun H., Weisbrod J. Evolving Feedforward Neural Networks. Proc. of the 1993 International Conference on Artificial Neural Nets and Genetic Algorithms

[5] Griffith N. and Todd P. M, editors. Musical Networks. MIT Press, 1997.

[6] URL: http://www.jardindegente.com.ar/index.php?nota=prensa_365_1. Página válida al 20/09/2011.

[7] Spector, L., and Alpern A. 1994. Criticism, Culture, and the Automatic Generation of Artworks. In Proceedings of the Twelfth National Conference on Artificial Intelligence, AAAI-94, pp. 3-8. Menlo Park, CA and Cambridge, MA: AAAI Press/The MIT Press.

[8] Todd, P. M. & Werner, G. M. (1999). Frankensteinian Methods for Evolutionary Music Composition. In Griffith and Todd, P. M. (Eds.) Musical networks: Parallel perception and performance, 313-339.

[9] Todd, P.M. (1988) A sequential network design for musical applications. In D. Touretzky, G. Hinton, & T. Sejnowski (Eds.), Proceedings of the 1988 Connectionist Models Summer School (pp. 76-84). San Mateo, CA: Morgan Kaufmann.

[10] Todd, P.M. (1989) A connectionist approach to algorithmic composition. Computer Music Journal, 13(4), 27-43.

[11] Todd S. & Latham, W. (1992) Evolutionary Art and Computers. New York: Academic Press.

[12] Wiggins G., Papadopoulos G., Phon-Amnuaisuk S., and Tuson A.. Evolutionary Methods for Musical Composition. International Journal of Computing Anticipatory Systems (inpress), 1999.

[13] Bharucha, J.J. & Todd, P.M. (1989) Modeling the perception of tonal structure with neural nets. Computer Music Journal, 13(4), 44-53. Also in P.M. Todd & D.G. Loy (Eds.), Music and Connectionism (pp. 128-137). Cambridge, MA: MIT Press.

[14] Burton A. and Vladimirova T.. A Genetic Algorithm Utilising Neural Network Fitness Evaluation for Musical Composition. In International Conference on Genetic Algorithms and Artificial Neural Networks, 1997a.

[15] Burton A. R. & Vladimirova T. (1999). Generation of Musical Sequences with Genetic Techniques. Computer Music Journal, 23 (4), 59-73.

[16] Burton, A. and Vladimirova, T. (1997). Applications of Genetic Techniques to Musical Composition.

[17] Feulner J., Hörnel D.. MELONET: Neural Networks that Learn Harmony Based Melodic Variations. Proc. of the 1994 International Computer Music Conference, ICMA Århus 1994

[18] Feulner J., Neural Networks that Learn and Reproduce Various Styles of Harmonization. Proc. of the 1993 International Computer Music Conference, ICMA Tokyo 1993

[19] Giomi F. and Ligabue M. Computational Generation and Study of Jazz.

[20] Griffith N. and Todd P.M. (Eds.) (1998). Musical networks: Parallel distributed perception and performance. Cambridge, MA: MIT Press/Bradford Books.

[21] Hörnel, D., & Ragg, T. (1996) Learning musical structure and style by recognition, prediction and evolution. In Proceedings of the 1996 International Computer Music Conference (pp. 59-62). San Francisco: International Computer Music Association.

[22] Horner, A. & Goldberg, D. E. (1991). Genetic algorithms and Computer Assisted Music Composition. ICMC'91 Proceedings, San Francisco: International Computer Music Association, 479-482.

[23] Horner, A. and Goldberg, D. (1991). Genetic algorithms and computer-assisted composition. In Proceedings of the Fourth International Conference on Genetic Algorithms.

[24] Horner, A., & Ayers, L. (1995) Harmonization of musical progressions with genetic algorithms. In Proceedings of the 1995 International Computer Music Conference (pp. 483-484). San Francisco: International Computer Music Association.

[25] Horner, A., Assad, A., & Packard, N. (1994) Artificial music: The evolution of musical strata. Leonardo Music Journal, 3, 81.

[26] Horowitz, D. (1994). Generating rhythms with genetic algorithms. In Proceedings of the International Computer Music Conference.

[27] Koza, J. (1993) Genetic Programming. Cambridge, MA: MIT Press/Bradford Books.

[28] Moore, J.H. (1994) GAMusic: Genetic algorithm to evolve musical melodies.

[29] Moroni A., Manzoulli J., Von Zuben F., Gudwin R., Vox Populi: Evolutionary computation for music evolution. In P.J. Bentley, D. W. Corne, Creative evolutionary Systems, Morgan Kaufmann, 2002, pp. 205-221.

[30] Putnam, J.B. (1994) Genetic Programming of Music. Unpublished manuscript. Socorro, NM: New Mexico Institute of Mining and Technology.

[31] Ralley D. (1995). Genetic algorithms as a tool for melodic development. In Proceedings of the 1995

[32] Ralley D. Genetic algorithms as a tool for melodic development. In Proceedings of the International Computer Music Conference, 1995.

[33] Thywissen, K. (1996) GeNotator: An environment for investigating the application of genetic algorithms in computer assisted composition. In Proceedings of the 1996 International Computer Music Conference. San Francisco: International Computer Music Association.

[34] Towsey, M., Brown, A., Wright, S., & Diederich, J. (2000). Interactive Music Composition using Genetic Algorithms. Paper presented at the 7thConf. of the International Society for the Study of European Ideas (ISSEI), 14-18 August, 2000, University of Bergen, Norway.

[35] Gibson, P. and Byrne, J. (1991). Neurogen, musical composition using genetic algorithms and cooperating neural networks. In Proceedings of the 2nd International Conference in Artificial Neural Networks.

[36] Jacob, B. (1995). Composing with genetic algorithms. In Proceedings of the International Computer Music Conference.

[37] Biles, J.A. (1994). Genjam: A genetic algorithm for generating jazz solos. In Proceedings of the International Computer Music Conference.

[38] Biles J.A, Autonomous GenJam: Eliminating the Fitness Bottleneck by Eliminating Fitness, Workshop on Non-routine Design with Evolutionary Systems, Genetic and Evolutionary Computation Conference, 2001.

[39] Biles, J., Anderson, P., and Loggi, L. (1996). Neural network fitness functions for a musical IGA. Technical report, Rochester Institute of Technology.

[40] Biles, J.A. (1995a) The AI Biles Virtual Quintet: GenJam. Compact disc recording DRK-CD-144. Rochester, NY: Dynamic Recording Studios.

[41] Biles, J.A. (1995b) GenJam Populi: Training an IGA via audience-mediated performance. In Proceedings of the 1995 International Computer Music Conference (pp. 347-348). San Francisco: International Computer Music Association.



Ezequiel Moldaver. Es Ingeniero en Informática por la Facultad de Ingeniería de la Universidad de Buenos Aires. Es docente de física I en la misma facultad y actualmente trabaja como desarrollador de software.



Hernán Merlino. Es Magister en Ingeniería de Software por la Universidad Politécnica de Madrid. Es Profesor Titular Regular de la Licenciatura en Sistemas y Director del Laboratorio de Investigación y Desarrollo en Arquitecturas Complejas de la Universidad Nacional de Lanús. Es Profesor Adjunto Regular de la Cátedra de Sistemas de Programación no Convencional de Robots de la Facultad de Ingeniería de la Universidad de Buenos Aires. Es Docente Investigador del Programa de Incentivos de la Secretaría de Políticas Universitarias del Ministerio de Educación. Su áreas de investigación son: patrones de diseño y de desarrollo de software; sistemas inteligentes y sus aplicaciones.



Enrique Fernández. Es Doctor en Ciencias Informáticas por la Universidad Nacional de La Plata. Es Investigador Adscripto al Grupo de Investigación en Sistemas de Información de la Licenciatura en Sistemas de la Universidad Nacional de Lanús. Es Docente de la Cátedra de Sistemas de Programación no Convencional de Robots de la Facultad de Ingeniería de la Universidad de Buenos Aires..