
UNIVERSIDAD AUTÓNOMA DE GUADALAJARA

CON RECONOCIMIENTO DE VALIDEZ OFICIAL DE ESTUDIOS DE LA SECRETARÍA DE
EDUCACIÓN PÚBLICA SEGÚN ACUERDO No.158 DE FECHA 17 DE JULIO DE 1991

POSTGRADO E INVESTIGACIÓN



Uso de Deep Learning para composición musical

TRABAJO DE INVESTIGACIÓN

QUE PRESENTA

EFRAIN ADRIAN LUNA NEVAREZ

PARA OBTENER EL GRADO DE

MAESTRÍA EN CIENCIAS COMPUTACIONALES

DIRECTOR Y ASESOR:

MTRO. JUAN ANTONIO VEGA FERNANDEZ

GUADALAJARA, JALISCO. AGOSTO 2018

Número de registro en el sistema de investigación

Tabla de Contenido

Lista de Figuras	III
Lista de Tablas	v
1. Introducción	1
1.1. Descripción del Problema	1
1.2. Objetivos	1
1.2.1. Objetivo General	1
1.2.2. Objetivos Específicos	2
1.3. Justificación	2
1.4. Delimitación	2
1.5. Organización de la Tesis	3
2. Marco Teórico	5
2.1. Inteligencia Artificial	5
2.1.1. Redes neuronales artificiales	7
2.1.2. Deep Learning	11
2.1.3. Redes neuronales recurrentes	11
2.1.4. Composición musical	14
2.2. Teoría musical	14
2.2.1. Notas musicales	15
2.2.2. Claves musicales	17
2.2.3. Compases y tiempo	18
2.2.4. Escalas	20
2.2.5. Armaduras	21

2.2.6. Tonalidades	22
2.2.7. Melodías y Armonías	22
2.2.8. Formato MIDI	23
3. Desarrollo	27
3.1. Desarrollo del proyecto	27
3.1.1. Datos	27
3.1.2. Descripción del proyecto	28
3.1.3. Proceso de entrenamiento	29
3.1.4. Validación de resultados	29
4. Conclusiones	31
Bibliografía	33

Lista de Figuras

2.1. Ramas de la inteligencia artificial	5
2.2. Red neuronal biológica	7
2.3. Estructura general de una neurona	8
2.4. Neurona Artificial	8
2.5. Relacion de tonos y semitonos	16
2.6. Pentagrama	16
2.7. Clave de Sol en 2da. Linea	17
2.8. Clave de Fa en 4ta. Linea	17
2.9. Clave de Do en 4ta. y 3ra. linea	17
2.10. Nombre de las notas en un pentagrama con clave de Sol	18
2.11. Duración de las notas musicales	18
2.12. Valor relativo de las notas	19
2.13. Compas	19
2.14. Tiempos fuertes y débiles	19
2.15. Escala de Do Mayor	20
2.16. Escala de Sol Mayor	20
2.17. Escala de La menor	21
2.18. Armaduras relativas con sostenidos	21
2.19. Armaduras relativas con bemoles	22
2.20. Melodía y Armonía	23
2.21. Extracto de un formato MIDI	24
2.22. Partitura correspondiente al extracto MIDI	24
2.23. Instrumentos en formato MIDI	25
3.1. Estructura de la Red neuronal	28
3.2. Arquitectura de la aplicación	29

Lista de Tablas

Capítulo 1

Introducción

La música es el idioma universal, no importa en qué lugar físico nos encontremos, todos alguna vez hemos escuchado alguna canción que se nos queda grabada en la cabeza por mucho tiempo. En ocasiones nos preguntamos cómo es que el autor compuso esa canción.

Si bien es cierto el componer una canción es un arte, las canciones se pueden interpretar en forma matemática, esto nos abre la posibilidad de usar algoritmos para la creación de nuevas canciones.

1.1. Descripción del Problema

El tiempo requerido para la creación de una nueva canción es muy variado, ya que depende del género, el número de instrumentos y por supuesto la imaginación del compositor. Sin embargo si pudiéramos usar un algoritmo de aprendizaje en una computadora, esto nos permitiría generar nuevas canciones de una manera más sencilla y tomando mucho menos tiempo.

Existen compositores que a lo largo del tiempo no les gusta explorar nuevas formas o ritmos musicales y eso ocasiona que sus composiciones sean muy similares entre si.

1.2. Objetivos

1.2.1. Objetivo General

Creación de un programa que use algoritmos para servir de apoyo para composición musical.

1.2.2. Objetivos Específicos

- 1.- Implementar una red neuronal de Deep Learning que sea capaz de crear melodías y armonías musicales en guitarra.
- 2.- Implementar una red neuronal de Deep Learning que sea capaz de crear melodías musicales en bajo eléctrico.
- 3.- Conjuntrar la salida de las redes neuronales para generar un solo archivo de audio.
- 4.- Verificar la Tonalidad de la canción resultante.

1.3. Justificación

El uso de algoritmos de Deep Learning en la música no es algo nuevo, sin embargo, existen muchas áreas de oportunidad en cuestión de la implementación de estos algoritmos.

La mayoría de trabajos en esta área se basan en sonidos melódicos y no abarcan tan a fondo el campo armónico. Lo que se pretende en esta investigación es tener una manera fácil y rápida para la creación de nuevas composiciones musicales partiendo ya sea por melodías o armonías musicales de uno o varios instrumentos.

Un algoritmo eficaz para composición es aquel que es capaz de aprender a componer música partiendo de tres fundamentos musicales: la tonalidad, el ritmo y la cuadratura, el algoritmo que se pretende realizar en esta investigación evaluará estas variables y las tratará de generalizar para poder crear música a partir de una simple melodía o una armonía.

1.4. Delimitación

Este proyecto se centrará en crear melodías musicales de guitarra y bajo eléctrico así como también ritmos básicos de batería, no se analizarán otros instrumentos.

Las nuevas canciones creadas por estos algoritmos no buscan ser ideas finales de composición, la única intención es crear una serie de ideas para nuevas canciones.

La base de datos usada para entrenar a la red neuronal consta de canciones de género rock y pop unicamente, no se entrenara la red para reconocer y generar canciones de otros géneros musicales.

1.5. Organización de la Tesis

A continuación se muestra la descripción de cada capítulo:

- **Capítulo 1.-** En el primer capítulo se ve la descripción del problema, objetivos generales y justificación del proyecto.
- **Capítulo 2.-** Este capítulo contiene marco teórico del proyecto, en el cual se describen los conceptos generales usados en el proyecto.
- **Capítulo 3.-** En este capítulo se abordara el desarrollo del proyecto así como los resultados obtenidos.
- **Capítulo 4.-** En este último capítulo se ven las conclusiones del proyecto.

Capítulo 2

Marco Teórico

2.1. Inteligencia Artificial

En computación el término Inteligencia Artificial se define como la facultad de razonamiento de un programa informático, este agente racional tiene la capacidad de percibir su entorno y lleva a cabo acciones para maximizar el éxito en una tarea asignada.

Cuando una maquina es capaz de imitar las funciones cognitivas del ser humano para aprender y resolver problemas se puede decir que la maquina posee inteligencia artificial.

Dentro de la inteligencia artificial tenemos varias ramas que se enfocan en la resolución de problemas aplicando un principio muy específico de la inteligencia.

Las ramas de la inteligencia artificial se pueden dividir en áreas clásicas y áreas de vanguardia de acuerdo con la época cuando surgieron:

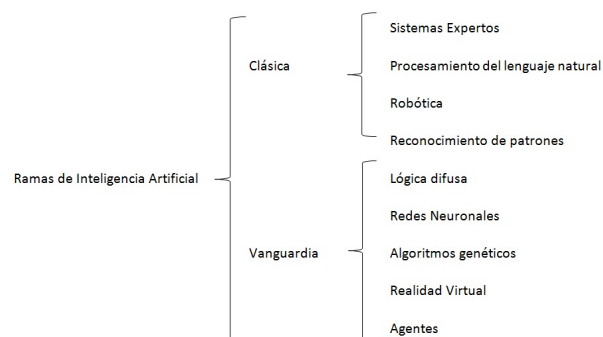


Figura 2.1: Ramas de la inteligencia artificial

Sistemas expertos.- son conocidos también como sistemas de conocimientos y estos programas informáticos aplican el proceso de razonamiento de un humano exper-

to en la materia en la solución de problemas específicos. El modo de procesamiento de estos sistemas es en base a una gran base de datos y utilizando una heurística avanzada para la determinación de las posibles soluciones a un problema.

Procesamiento del lenguaje natural.- estos son sistemas capaces de reconocer, procesar y en cierto punto emular la comunicación humana, estos sistemas buscan dejar el uso de lenguajes de programación o conjunto de comandos, para procesar el lenguaje humano natural. Para procesar el lenguaje es necesario dividirlo, primero se obtiene la comprensión del lenguaje natural, el cual investiga los métodos para que una computadora sea capaz de entender las instrucciones de este lenguaje, la segunda etapa consiste en la generación de lenguaje natural, aquí es donde la maquina intenta comunicarse en el lenguaje humano.

Robótica.- un robot es un dispositivo programado para realizar una tarea en específico. Se dice que un robot está adquiriendo inteligencia artificial si es capaz de responder a cambios en su entorno en lugar de seguir instrucciones programadas con anterioridad.

Reconocimiento de patrones.- esta es la parte de la inteligencia artificial encargada del procesamiento visual de un entorno, las imágenes son captadas por cámaras y posteriormente procesadas para el reconocimiento de patrones del entorno.

Lógica difusa.- es una forma matemática de representar el lenguaje natural, y el principio es generalizar la lógica clásica haciendo que las variables tomen valores lingüísticos de verdad.

Redes neuronales.- estas redes tratan de emular el comportamiento de las redes neuronales biológicas que poseen los humanos en su cerebro, partiendo de una neurona artificial y conectándola con otras para crear sinapsis entre ellas.

Algoritmos genéticos.- estos algoritmos son capaces de mutar para producir mejores respuestas a un entorno, estos sistemas tratan de imitar el proceso de selección natural en el cual al ir mutando algunos genes se van obteniendo sistemas más capaces para un entorno. Su función es seleccionar de una población de soluciones candidatas e intentar producir nuevas generaciones de soluciones las cuales se buscan que sean mejores que las anteriores.

Realidad virtual.- es la recreación de un mundo artificial en tiempo real que puede ser captado por distintos canales sensoriales del espectador. Agentes.- estos son pequeños programas que actúan como espías observando las acciones comúnmente realizadas por el usuario, estas acciones son almacenadas y registradas, si en dado

caso se llega a dar una anomalía el programa lanzara una alerta y dará una serie de soluciones.

2.1.1. Redes neuronales artificiales

Las redes neuronales artificiales son un conjunto de neuronas creadas artificialmente para el desarrollo de inteligencia artificial en una computadora.

Estas redes están basadas en las redes biológicas del cerebro humano, modelando todos los factores biológicos de las neuronas.



"Jiménez, J., Pensamientos habituales y desarrollo personal. [Red neuronal]. Recuperado de <http://poderpersonalmexico.com/tag/redes-neuronales>"

Figura 2.2: Red neuronal biológica

Debido a su diseño las redes son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos, de abstraer características esenciales a partir de entradas que representan en ocasiones información irreverente.

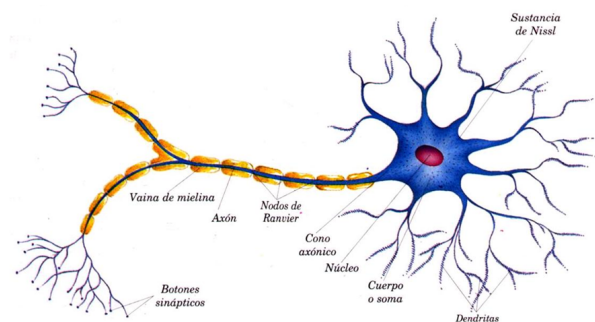
La capacidad de aprendizaje adaptativo es una característica fundamental de las redes neuronales y les permiten llevar a cabo ciertas tareas mediante un entrenamiento previo, pueden aprender a diferenciar patrones y generalizar a partir de estos. Son considerados sistemas dinámicos ya que son capaces de adaptarse a nuevas condiciones de entrada.

Tienen una alta tolerancia a fallos ya que son capaces de detectar patrones aun cuando estos patrones posean ruido, distorsión o simplemente están incompletos. Estos programas son capaces de seguir funcionando incluso si parte de la red presente fallas.

La información se almacena de forma distribuida en las conexiones de las neuronas, provocando redundancia de información, es decir se guardara sus valores en base a

la función de activación que posee cada neurona, de esta manera si una neurona es destruida o presenta fallas, las otras neuronas podrán aprender la información de la neurona que fallo.

Las neuronas humanas poseen diferentes secciones:



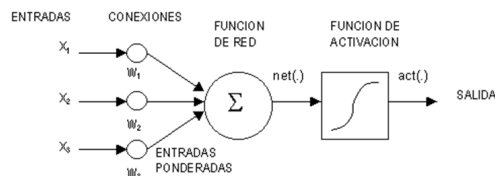
"Autor anónimo, Cuadros sinópticos y mapas conceptuales sobre las neuronas [Estructura general de una neurona]. Recuperado de <http://cuadrocomparativo.org/cuadros-sinopticos-sobre-las-neuronas>"

Figura 2.3: Estructura general de una neurona

En una neurona artificial se busca la emulación de las principales secciones de una neurona las cuales son:

- **Cuerpo.-** Se encarga de producir un impulso eléctrico en base a las entradas de la neurona.
- **Dendritas.-** son filamentos capaces de crear conexiones con otras neuronas.
- **Axón.-** Es el encargado de transmitir el impulso eléctrico generado por el cuerpo.

A continuación se muestra una imagen de como luciría una neurona artificial:



"Autor anónimo, Introducción a las redes neuronales artificiales [Neuronal artificial]. Recuperado de <http://www.rma.30webs.com/>"

Figura 2.4: Neurona Artificial

Esta neurona posee las siguientes secciones:

- **X1, X2, ..., Xn**.- Son las entradas de la neurona.
- **W1, W2, ..., Wn**.- Pesos específicos que tendrá cada entrada, esto hace que las entradas no valgan lo mismo ponderadamente.
- **Función de red**.- Esta es una función de sumatoria de las entradas.
- **Función de activación**.- Si la suma de las entradas es mayor o igual que el umbral definido por esta función se tendrá una señal a la salida.

La salida de la neurona viene dada por esta ecuación:

$$y_j = f\left(\sum_{i=1}^n (W_{ij}x_i + \theta_j)\right) \quad (2.1.1)$$

Una red neuronal no es más que la interconexión de varias neuronas artificiales, en la cual podemos identificar al menos tres secciones:

- **Capa de entrada**.- En esta capa se procesan todas las entradas, y si estas entradas son capaces de excitar las neuronas de esta capa se producirá una señal de salida.
- **Capa oculta**.- En esta capa se encuentran las neuronas encargadas del aprendizaje de la red.
- **Capa de salida**.- Esta neurona o neuronas de salida tendrán la salida del sistema.

La forma en que las redes aprenden es mediante la modificación de los pesos de las entradas.

Las redes neuronales artificiales han ido evolucionando con el paso del tiempo, hoy en día existen muchos modelos de redes neuronales, todas ellas con ventajas y desventajas si son comparadas entre ellas.

Aprendizaje de las redes neuronales

El procedimiento utilizado para llevar a cabo el proceso de aprendizaje en una red neuronal se denomina entrenamiento.

El problema de aprendizaje en las redes neuronales se formula en términos de la minimización de la función de error (o pérdida) asociada.

Normalmente, esta función está compuesta por dos términos, uno que evalúa cómo se ajusta la salida de la red neuronal al conjunto de datos de que disponemos, y que se denomina término de error, y otro que se denomina término de regularización, y que se utiliza para evitar el sobreaprendizaje por medio del control de la complejidad efectiva de la red neuronal.

Por supuesto, el valor de la función de error depende por completo de los parámetros de la red neuronal: los pesos sinápticos entre neuronas, y los bias asociados a ellas, que, como suele ser ya habitual, se pueden agrupar adecuadamente en un único vector de peso de la dimensión adecuada, que denotaremos por w . En este sentido, podemos escribir $f(w)$ para indicar que el valor del error que comete la red neuronal depende de los pesos asociados a la misma. Con esta formalización, nuestro objetivo es encontrar el valor w^* para el que se obtiene un mínimo global de la función f , convirtiendo el problema de aprendizaje en un problema de optimización.

En general, la función de error es una función no lineal, por lo que no disponemos de algoritmos sencillos y exactos para encontrar sus mínimos. En consecuencia, tendremos que hacer uso de una búsqueda a través del espacio de parámetros que, idealmente, se aproxime de forma iterada a un (error) mínimo de la red para los parámetros adecuados.

De esta forma, se comienza con una red neuronal con algún vector inicial de parámetros (a menudo elegido al azar), a continuación se genera un nuevo vector de parámetros, esperando que con ellos la función de error se reduzca (aunque dependiendo del método elegido, no es obligatorio, y temporalmente se puede admitir un empeoramiento del error siempre y cuando conduzca a una disminución posterior más acusada). Este proceso se repite, normalmente, hasta haber reducido el error bajo un umbral tolerable, o cuando se satisfaga una condición específica de parada.

El Descenso del Gradiente es el algoritmo de entrenamiento más simple y también el más extendido y conocido. Solo hace uso del vector gradiente, y por ello se dice que es un método de primer orden.

Este método para construir el punto w_{i+1} a partir de w_i se traslada este punto en la dirección de entrenamiento $d_i = -g_i$. Es decir:

$$w_{i+1} = w_i - g_i v_i \quad (2.1.2)$$

Donde el parámetro v se denomina tasa de entrenamiento, que puede fijarse a priori o calcularse mediante un proceso de optimización unidimensional a lo largo de la dirección de entrenamiento para cada uno de los pasos (aunque esta última opción es preferible, a menudo se usa un valor fijo, $v_i = v$ con el fin de simplificar el proceso).

Aunque es muy sencillo, este algoritmo tiene el gran inconveniente de que, para funciones de error con estructuras con valles largos y estrechos, requiere muchas iteraciones. Se debe a que, aunque la dirección elegida es en la que la función de error disminuye más rápidamente, esto no significa que necesariamente produzca la convergencia más rápida.

Por ello, es el algoritmo recomendado cuando tenemos redes neuronales muy grandes, con muchos miles de parámetros, ya que sólo almacena el vector gradiente (de tamaño n).

2.1.2. Deep Learning

Deep Learning usa redes neuronales con muchas capas para lograr aprendizajes más complejos. Este comportamiento asemeja la forma en que el cerebro humano toma decisiones, el cual usa la interconexión de varias capas de neuronas para realizar actividades complejas. Dentro de las redes de Deep Learning se tienen 2 tipos muy usados en la actualidad:

- **Redes convolucionales (CNN).**- Este tipo de redes usa la convolución en varias de sus capas para lograr el procesamiento de parámetros que se pueden representar en un espacio R^2 , un ejemplo claro de esto son las imágenes y vídeos, por lo tanto si se quiere hacer una clasificación o reconocimiento de imágenes, este tipo de redes nos proporcionan una buena herramienta de procesamiento.
- **Redes recurrentes (RNN).**- Este tipo de redes son muy usadas cuando se busca analizar una secuencia de datos, estas redes poseen memoria y una retroalimentación de la salida a la entrada.

2.1.3. Redes neuronales recurrentes

La idea detrás de las RNN es hacer uso de la información secuencial. En una red neuronal tradicional suponemos que todas las entradas (y salidas) son independientes entre sí. Pero para muchas tareas eso es una muy mala idea. Si quieres predecir la

siguiente palabra en una oración, es mejor que conozcas qué palabras vienen antes. Las RNN se llaman recurrentes porque realizan la misma tarea para cada elemento de una secuencia, y la salida depende de los cálculos previos. Otra forma de pensar acerca de las RNN es que tienen una "memoria" que captura información sobre lo que se ha calculado hasta ahora. En teoría, los RNN pueden hacer uso de la información en secuencias arbitrariamente largas, pero en la práctica se limitan a mirar hacia atrás solo unos pocos pasos.

La decisión de una red recurrente alcanzada en el paso de tiempo $t-1$ afecta la decisión que alcanzará un momento más tarde en el paso de tiempo t . Entonces, las redes recurrentes tienen dos fuentes de entrada, el presente y el pasado reciente, que se combinan para determinar cómo responden a los datos nuevos, de forma similar a como lo hacemos en la vida.

Esa información secuencial se conserva en el estado oculto de la red recurrente, que logra abarcar muchos pasos de tiempo a medida que avanza para afectar el procesamiento de cada nuevo ejemplo. Está encontrando correlaciones entre eventos separados por muchos momentos, y estas correlaciones se llaman "dependencias a largo plazo", porque un evento en el tiempo depende de, y es una función de, uno o más eventos que vinieron antes. Una forma de pensar acerca de las RNN es esta: son una forma de compartir pesos a lo largo del tiempo.

Así como la memoria humana circula invisiblemente dentro de un cuerpo, afectando nuestro comportamiento sin revelar su forma completa, la información circula en los estados ocultos de las redes recurrentes.

Describiremos el proceso de llevar la memoria hacia adelante matemáticamente:

$$h_t = \phi(Wx_t + Uh_{t-1}) \quad (2.1.3)$$

El estado oculto en el paso de tiempo t es h_t . Es una función de la entrada al mismo tiempo x_t , modificada por una matriz de ponderación W (como la que usamos para las redes feedforward) agregada al estado oculto del paso de tiempo anterior h_{t-1} multiplicado por su propio estado oculto matriz U de estado oculto, también conocida como matriz de transición y similar a una cadena de Markov. Las matrices de peso son filtros que determinan la importancia de acuerdo tanto con la entrada actual como con el estado oculto pasado. El error que generan volverá a través de la propagación inversa y se usará para ajustar sus ponderaciones hasta que el error no pueda bajar más.

Debido a que este ciclo de retroalimentación ocurre en cada paso de la serie, cada estado oculto contiene rastros no solo del estado oculto anterior, sino también de todos los que precedieron a h_{t-1} mientras la memoria pueda persistir.

LSTM

A mediados de los años 90, los investigadores alemanes Sepp Hochreiter y Juergen Schmidhuber propusieron una variación de la red recurrente con las denominadas unidades de memoria a largo plazo, o LSTM, como una solución al problema del gradiente de fuga.

Los LSTM ayudan a preservar el error que se puede volver a propagar a través del tiempo y las capas. Al mantener un error más constante, permiten que las redes recurrentes continúen aprendiendo durante muchos pasos de tiempo (más de 1000), abriendo así un canal para vincular causas y efectos de forma remota. Este es uno de los desafíos centrales para el aprendizaje automático y la IA, ya que los algoritmos se enfrentan con frecuencia a entornos en los que las señales de recompensa son dispersas y diferidas, como la vida misma.

Los LSTM contienen información fuera del flujo normal de la red recurrente en una celda cerrada. La información puede almacenarse, escribirse o leerse desde una celda, al igual que los datos en la memoria de una computadora. La célula toma decisiones sobre qué almacenar y cuándo permitir las lecturas, escrituras y borraduras, a través de puertas que se abren y cierran. Sin embargo, a diferencia del almacenamiento digital en computadoras, estas puertas son análogas, implementadas con la multiplicación de elementos por sigmoides, que están todas en el rango de 0-1. Siendo Analógica tiene la ventaja sobre digital de ser diferenciable y, por lo tanto, adecuado para la propagación inversa.

Esas puertas actúan sobre las señales que reciben, y de forma similar a los nodos de la red neuronal, bloquean o transmiten información en función de su fuerza e importación, que filtran con sus propios conjuntos de ponderaciones. Esos pesos, como los pesos que modulan los estados de entrada y ocultos, se ajustan a través del proceso de aprendizaje de redes recurrentes. Es decir, las células aprenden cuándo permiten que los datos entren, salgan o se eliminen a través del proceso iterativo de hacer conjeturas, volver a propagar el error y ajustar los pesos mediante el descenso del gradiente.

2.2. Teoría musical

La teoría musical es el estudio de los elementos que conforman la música. En esta teoría se analizan todos los sonidos involucrados para la creación, análisis y composición musical.

La música es un arte que se basa en 2 elementos:

- Los sonidos
- Los silencios

Los sonidos tienen diferentes propiedades las cuales se describen a continuación:

- **Altura.-** un sonido puede ser agudo, medio o grave, dependiendo de la altura de su nota.
- **Duración.-** un sonido debe de tener una duración la cual se expresa en unidades de tiempo.
- **Intensidad.-** esto se refiere al volumen del sonido, puede ser débil o fuerte.
- **Timbre.-** se le conoce como timbre o color del sonido a como un sonido con la misma nota suena diferente dependiendo del instrumento usado para su interpretación.

Los silencios a su vez su única propiedad intrínseca es la duración, es decir en un silencio lo único que se mide es la duración del mismo.

2.2.1. Composición musical

La composición musical esta catalogado como un arte que tiene como objeto crear nuevas piezas musicales.

Un músico puede optar por varios caminos para la creación de su obra, existen músicos que se basan en su simple sentido común y crean canciones líricamente, sin embargo el proceso formal de composición involucra todos los conceptos musicales básicos de la teoría musical.

2.2.2. Notas musicales

Las notas es un sistema que se usa para la representación de los diferentes sonidos en la música. En el mundo occidental se usa un sistema de 12 notas, los cuales pueden ser repetidos con diferentes alturas para generar una gama muy amplia de sonidos. Dentro de este sistema de 12 notas tenemos las notas naturales y las notas con alteraciones. Las notas naturales son:

Do, Re, Mi, Fa, Sol, La, Si (7 notas)

Las alteraciones no son más que agregar o quitar medios tonos a una nota, para eso se usan los siguientes símbolos:

- **# (sostenido).**- se agrega $1/2$ tono a una nota.
- **x (doble sostenido).**- Se agrega 1 tono a una nota.
- **b (bemol).**- se disminuye $1/2$ tono a una nota.
- **bb (doble bemol).**- Se disminuye 1 tono a una nota.

Usando los símbolos anteriores podemos definir las siguientes notas con alteraciones:

Do#, Re#, Fa#, Sol#, La# (5 notas)

Como se puede observar tanto Mi y Si no tienen sonidos con alteraciones ascendentes, ya que los sonidos producidos por estas alteraciones son igual al de las notas consecutivas, a este tipo de sonidos iguales se les conoce como notas enarmónicas.

Por ejemplo:

- Mi# sonaría exactamente igual que un Fa.
- Si# sonaría exactamente igual que un Do.

También las notas con alteraciones pueden ser representadas usando el símbolo de bemol (b), es decir restando medio tono a una nota. En este caso se tendrían las notas con alteraciones de la siguiente manera:

Re_b, Mi_b, Sol_b, La_b, Si_b (5 notas)

Como se mencionó anteriormente las notas que en sonido son exactamente iguales pero en nomenclatura son diferentes se conocen como notas enarmónicas, de tal manera que las 5 notas con alteraciones que se describieron se pueden hacer una comparación del sonido de las mismas, siendo así se tiene:

- Do# suena exactamente igual que Re \flat .
- Re# suena exactamente igual que Mi \flat .
- Fa# suena exactamente igual que Sol \flat .
- Sol# suena exactamente igual que La \flat .
- La# suena exactamente igual que Si \flat .

Podemos ver la relación de tonos y semitonos (1/2 tonos) en la siguiente figura:

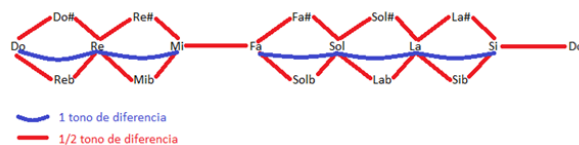


Figura 2.5: Relacion de tonos y semitonos

Como se puede observar entre Mi y Fa existe un semitono, al igual que entre Si y Do.

El pentagrama es un sistema de cinco líneas y cuatro espacios para escribir música:



Figura 2.6: Pentagrama

Este sistema puede tener diferentes elementos dentro de los principales tenemos:

- Clave.
- Compás.
- Armadura.
- Notas.

2.2.3. Claves musicales

Las claves musicales se usan para darle nombre y altura a las notas musicales dentro de un pentagrama.

Las claves más usadas en la música son: la clave de Sol, la clave de Fa y la clave de Do.

La clave de Sol se usa en instrumentos agudos como la guitarra, violín, entre otros. Esta clave normalmente se escribe empezando en la segunda línea del pentagrama para formar una especie de G. El hecho de que esta clave se escriba en la segunda línea establece que esa línea será llamada como el nombre de la clave, en este caso Sol:



Figura 2.7: Clave de Sol en 2da. Línea

La clave de Fa es usada en instrumentos más graves, tales como el contrabajo, el bajo, etc. Normalmente se escribe empezando en la cuarta línea, de tal manera que esta línea tomara el nombre de Fa:



Figura 2.8: Clave de Fa en 4ta. Línea

La clave de Do es usada comúnmente para las voces, y esta clave se escribe en diferentes líneas dependiendo del timbre de la voz, para un tenor se usa la clave de Do en 4ta línea, mientras que para un soprano normalmente la clave se usa en 3ra o 2da línea:

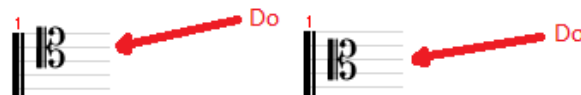


Figura 2.9: Clave de Do en 4ta. y 3ra. línea

A partir de estas claves se le puede poner nombre a las diferentes líneas y espacios del pentagrama, por ejemplo en la clave de sol en segunda línea, el pentagrama

quedaría de la siguiente forma:



Figura 2.10: Nombre de las notas en un pentagrama con clave de Sol

2.2.4. Compases y tiempo

El compás se puede definir como la unidad métrica de la música, es la que nos dice que tiempo llevara la canción.

Existen una infinidad de compases los cuales podemos clasificar en dos grandes grupos:

- Compases regulares.
- Compases irregulares.

Los compases regulares están regidos por formas regulares las cuales dictan el tiempo, mientras que en los compases irregulares hay que determinar la base de tiempo de forma indirecta.

Cada nota puede tener un valor de tiempo y este valor estará determinado por el compás que se esté utilizando, por ejemplo en un compás de 4/4 las figuras musicales tendrán los siguientes valores:

NOMBRE	FIGURA	SILENCIO	VALOR/PULSOS
Redonda			4 Tiempos
Blanca			2 Tiempos
Negra			1 Tiempo
Corchea			1 / 2 Tiempo
Semicorchea			1 / 4 Tiempo
Fusa			1/8 Tiempo
Semifusa			1/16 Tiempo

"Barraza, D. (2012), Figuras Musicales [Duración de las notas musicales]. Recuperado de <https://musicateoria.wordpress.com/figuras-musicales/>"

Figura 2.11: Duración de las notas musicales

Como se puede observar para cada figura musical existe su silencio correspondiente, el cual tendrá el mismo valor solo que este caso no se producirá sonido alguno.

Podemos tener también los valores relativos de estas notas respecto a otras notas:

		◊	♩	♩	♩	♩	♩	♩
REDONDA	◊	1	2	4	8	16	32	64
BLANCA	♩	$\frac{1}{2}$	1	2	4	8	16	32
NEGRA	♩	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4	8	16
CORCHEA	♩	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4	8
SEMICORCHEA	♩	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	1	2	4
FUSA	♩	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	1	2
SEMIFUSA	♩	$\frac{1}{64}$	$\frac{1}{32}$	$\frac{1}{16}$	$\frac{1}{8}$	$\frac{1}{4}$	$\frac{1}{2}$	1

"Goncalvez, T. (2015), *El valor de las notas musicales en tiempo* [Valor relativo de las notas]. Recuperado de <http://www.tucucu.com/2015/05/25/el-valor-de-las-notas-musicales-en-tiempo/>"

Figura 2.12: Valor relativo de las notas

De tal manera que en un compás tendremos dos datos los cuales se describen a continuación:



"Mari, Carmen (2011), *Lenguaje Musical [Compas]*. Recuperado de <https://mcarmenfer.wordpress.com/2011/01/07/compas/>"

Figura 2.13: Compas

Se puede ver que este compas tendrá 2 pulsos y el valor de cada pulso será de un tiempo de negra, ya que el valor relativo de la negra respecto a la redonda es de $1/4$.

Otro punto a considerar en los compases es que estos tienen tiempos fuertes, semifuertes y débiles. La cuadratura de una armonía usa estos tiempos para indicar los cambios que se deben de hacer.

Este es un ejemplo de los tiempos en compases de $4/4$, $3/4$ y $2/4$:



Figura 2.14: Tiempos fuertes y débiles

2.2.5. Escalas

Las escalas son una serie de notas musicales que siguen un orden establecido por intervalos desde una nota base. En el mundo de la música hay una infinidad de escalas pero todas parten de la escala mayor de Do:

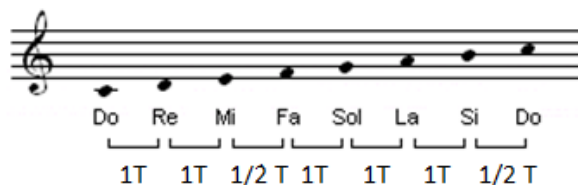


Figura 2.15: Escala de Do Mayor

En la escala mayor se tiene los siguientes intervalos: tono, tono, semitono, tono, tono, semitono, si esta fórmula la aplicamos con las otras notas musicales podremos construir todas las escalas mayores.

Por ejemplo la escala de sol mayor quedaría de la siguiente manera:

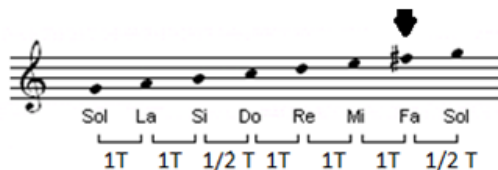


Figura 2.16: Escala de Sol Mayor

Para poder completar el tono completo de la fórmula de la escala mayor entre Mi y Fa se tuvo que poner una alteración en la nota de Fa.

La única escala mayor que no posee alteraciones es la escala mayor de Do, de ahí en más todas las demás escalas mayores tendrán al menos una alteración.

Las escalas menores parten de la escala mayor obteniendo su sexta nota y siguiendo las mismas notas de la escala mayor. Estas escalas menores se les conocen como menores relativas, ya que se basan en una escala mayor para su formación.

Por ejemplo la escala relativa de Do mayor sería La menor y esta escala posee exactamente las mismas notas que la escala mayor solamente que empieza desde la nota de La:

En este caso las notas son: La, Si, Do, Re, Mi, Fa, Sol, La.

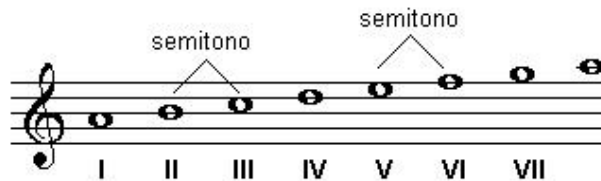


Figura 2.17: Escala de La menor

Esta relatividad puede ser usada con todas las escalas mayores para obtener sus relativas menores.

A pesar que las escalas mayores y menores poseen las mismas notas no deben ser nunca confundidas ya que el sonido final producido es muy diferente, mientras las escalas mayores se usan para canciones se puede decir hasta cierto punto alegres, las escalas menores normalmente acompañan melodías melancólicas, esto no es en todos los casos.

2.2.6. Armaduras

Todas las alteraciones de una escala se pueden juntar al inicio del pentagrama para dar lugar a lo que se conoce como armaduras.

Estas armaduras indicaran que todas las notas que poseen alteraciones las mantendrán a lo largo de toda la canción.

Debido a que las escalas mayores y sus relativas menores poseen las mismas alteraciones por lo tanto comparten también la misma armadura:

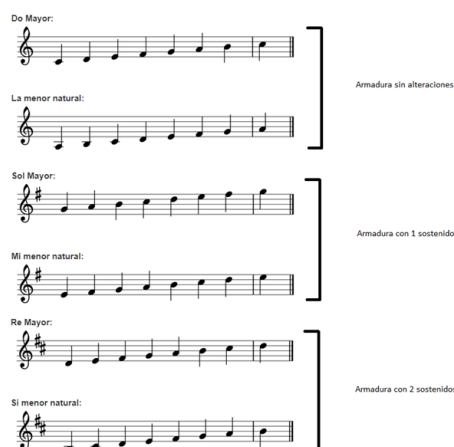


Figura 2.18: Armaduras relativas con sostenidos

También podemos tener armaduras con bemoles:

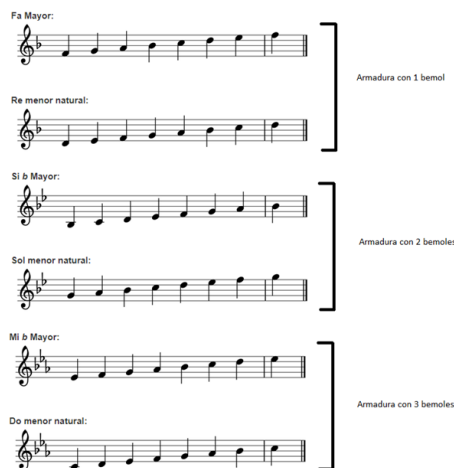


Figura 2.19: Armaduras relativas con bemoles

2.2.7. Tonalidades

La tonalidad de una canción se basa en la escala base de la canción y esta la podemos averiguar viendo la armadura que posee la canción, aunque como se vio anteriormente las escalas mayores y sus relativas poseen la misma armadura, así que antes de definir la tonalidad en base a la armadura también se debe de hacer un análisis de la interacción de las notas en la canción.

Dentro de las tonalidades tenemos tonalidades mayores y menores, por ejemplo si vemos un pentagrama el cual no posee alteraciones podríamos asumir que la canción esta en Do mayor o en La menor, el siguiente paso sería ver la interacción de las notas en la canción para determinar correctamente la tonalidad.

2.2.8. Melodías y Armonías

Una melodía es una sucesión de notas de forma ascendente o descendente que llevan cierta cordura.

Las melodías suelen estar formadas por frases y generalmente se repiten a lo largo de una canción variando algunas notas intermedias. En este caso se trata de una sucesión de notas que no son tocadas al mismo tiempo sino que una nota es tocada después de la otra.

Las armonías son una conjunción de sonidos tocados al mismo tiempo, normalmente la sucesión de varios acordes armónicos está ligado directamente con la melodía de la canción.

La armonía ha cambiado considerablemente desde la época de la música barroca hasta la música moderna, anteriormente para generar sistemas armónicos se usaban varios instrumentos tocando una nota en específico y la conjunción de todos los sonidos daba como resultado la armonía, actualmente la armonía es creada a partir de acordes de instrumentos que puedan generar este tipo de condiciones.

Esta es una comparativa para diferenciar entre lo que sería una melodía y una armonía:



Figura 2.20: Melodía y Armonía

2.2.9. Formato MIDI

MIDI (Interfaz digital de instrumentos musicales) es un estándar técnico que describe un protocolo, una interfaz digital y conectores para la interoperabilidad entre varios instrumentos musicales electrónicos, software y dispositivos. MIDI transmite mensajes de eventos que especifican información de notas (como tono y velocidad), así como señales de control para parámetros (como volumen, vibrato y señales de reloj). Hay cinco tipos de mensajes y aquí solo consideramos el tipo de Canal de voz, que transmite datos de rendimiento en tiempo real a través de un solo canal. Dos mensajes importantes son:

- **Note on.-** Indica que una nota debe ser reproducida. Contiene la información de estado (número de canal, especificado por un entero dentro de [0 15] y dos valores de datos: un número de nota MIDI (el tono de la nota, un entero dentro de [0 127]) y una velocidad (que indica cómo la nota es reproducida, un entero dentro de [0 127]). Un ejemplo es ¡Note on, 0, 60, 50! que interpreta como: ^{en} el canal 1, comienza a reproducir un C medio con una velocidad de 50".

- **Note off.-** Indica que una nota termina. En esa situación, la velocidad indica qué tan rápido se libera la nota. Un ejemplo es ¡Note off, 0, 60, 20! que interpreta como: .^{en} el canal 1, deja de reproducir un C medio con una velocidad de 20".

Cada evento de nota está realmente incrustado en un fragmento de pista, una estructura de datos que contiene un valor de tiempo delta que especifica la información de temporización y el evento en sí. Un valor de tiempo delta representa la posición de tiempo, como un valor absoluto, del evento y podría representar:

- **Metrical time.-** Representa el numero de pulsos desde el comienzo. Una referencia llamada división y es definida en el encabezado del archivo, especificando cuantos pulsos por nota de cuarto.
- **time-code-based time.-** Representa el tiempo relacionado con horas, minutos y segundos de la canción.

Un ejemplo de un extracto de un archivo MIDI y su partitura correspondiente es mostrado en las siguientes figuras. La división ah sido puesta en 384 pulsos por cuarto de nota lo cual corresponde a 96 pulsos por un octavo de nota:

```
2, 96, Note_on_c, 0, 60, 90
2, 192, Note_off_c, 0, 60, 0
2, 192, Note_on_c, 0, 62, 90
2, 288, Note_off_c, 0, 62, 0
2, 288, Note_on_c, 0, 64, 90
2, 384, Note_off_c, 0, 64, 0
2, 384, Note_on_c, 0, 65, 90
2, 480, Note_off_c, 0, 65, 0
2, 480, Note_on_c, 0, 62, 90
2, 576, Note_off_c, 0, 62, 0
```

Figura 2.21: Extracto de un formato MIDI



Figura 2.22: Partitura correspondiente al extracto MIDI

Dentro del formato MIDI se pueden representar en cada track un instrumento diferente eh independiente de los demás, esto permite una fácil manipulación de los

diferentes instrumentos. A continuación se muestra una tabla de todos los instrumentos que pueden ser representados en este formato:

PIANO	CHROM. PERCUS.	ORGAN	GIUITAR
01 - Acoustic Grand 02 - Bright Acoustic 03 - Electric Grand 04 - Honky-Tonk 05 - Electric Piano 1 06 - Electric Piano 2 07 - Harpsichord 08 - Clavinét	09 - Celesta 10 - Glockenspiel 11 - Music Box 12 - Vibraphone 13 - Marimba 14 - Xylophone 15 - Tubular Bells 16 - Dulcimer	17 - Drawbar Organ 18 - Percussive Organ 19 - Rock Organ 20 - Church Organ 21 - Reed Organ 22 - Accoridan 23 - Harmonica 24 - Tango Accordion	25 - Nylon String Guitar 26 - Steel String Guitar 27 - Electric Jazz Guitar 28 - Electric Clean Guitar 29 - Electric Muted Guitar 30 - Overdriven Guitar 31 - Distortion Guitar 32 - Guitar Harmonics
BASS	SOLO STRINGS	ENSEMBLE	BRASS
33 - Acoustic Bass 34 - Electric Bass(finger) 35 - Electric Bass(pick) 36 - Fretless Bass 37 - Slap Bass 1 38 - Slap Bass 2 39 - Synth Bass 1 40 - Synth Bass 2	41 - Violin 42 - Viola 43 - Cello 44 - Contrabass 45 - Tremolo Strings 46 - Pizzicato Strings 47 - Orchestral Strings 48 - Timpani	49 - String Ensemble 1 50 - String Ensemble 2 51 - SynthStrings 1 52 - SynthStrings 2 53 - Choir Aahs 54 - Voice Oohs 55 - Synth Voice 56 - Orchestra Hit	57 - Trumpet 58 - Trombone 59 - Tuba 60 - Muted Trumpet 61 - French Horn 62 - Brass Section 63 - SynthBrass 1 64 - SynthBrass 2
REED	PIPE	SYNTH LEAD	SYNTH PAD
65 - Soprano Sax 66 - Alto Sax 67 - Tenor Sax 68 - Baritone Sax 69 - Oboe 70 - English Horn 71 - Bassoon 72 - Clarinet	73 - Piccolo 74 - Flute 75 - Recorder 76 - Pan Flute 77 - Blown Bottle 78 - Skakuhachi 79 - Whistle 80 - Ocarina	81 - Square Wave 82 - Saw Wave 83 - Syn. Calliope 84 - Chiffer Lead 85 - Charang 86 - Solo Vox 87 - 5th Saw Wave 88 - Bass& Lead	89 - Fantasia 90 - Warm Pad 91 - Polysynth 92 - Space Voice 93 - Bowed Glass 94 - Metal Pad 95 - Halo Pad 96 - Sweep Pad
SYNTH EFFECTS	ETHNIC	PERCUSSIVE	SOUND EFFECTS
97 - Ice Rain 98 - Soundtrack 99 - Crystal 100 - Atmosphere 101 - Brightness 102 - Goblin 103 - Echo Drops 104 - Star Theme	105 - Sitar 106 - Banjo 107 - Shamisen 108 - Koto 109 - Kalimba 110 - Bagpipe 111 - Fiddle 112 - Shanai	113 - Tinkle Bell 114 - Agogo 115 - Steel Drums 116 - Woodblock 117 - Taiko Drum 118 - Melodic Tom 119 - Synth Drum 120 - Reverse Cymbal	121 - Guitar Fret Noise 122 - Breath Noise 123 - Seashore 124 - Bird Tweet 125 - Telephone Ring 126 - Helicopter 127 - Applause 128 - Gunshot

"María Quintanilla (2010). Protocolo general MIDI [Instrumentos en formato MIDI]. Recuperado de <http://cpms-acusticamusical.blogspot.com/2010/05/>"

Figura 2.23: Instrumentos en formato MIDI

Capítulo 3

Desarrollo

3.1. Desarrollo del proyecto

3.1.1. Datos

La base de datos de archivos midi que se utilizo en este proyecto se llama "Clean MIDI subset" la cual se uso en el proyecto [1]. Esta base de datos contiene más de 17,000 canciones en formato MIDI, estas canciones son en su mayoría del género Pop y Rock, por lo que la red utilizada en este proyecto fue entrenada para asimilar este tipo de estilos.

Para el procesamiento de los archivos MIDI se utilizo una librería de software libre llamada Music21, la cual nos permite de una manera fácil trabajar con estos archivos. Esta librería es compatible con Python3 por lo que nuestro programa esta hecho en este lenguaje de programación.

Delimitación de los Datos

Esta es una base de datos demasiada extensa por lo que el tiempo de procesamiento puede ser excesivo si se introducen los datos directamente a una red neuronal, debido a esto se decidió dividirla y solamente usar una parte de ella (1000 canciones unicamente).

Cada canción tiene una duración promedio de 3 minutos por lo que aunque se este usando solo una parte de las canciones la cantidad de datos es suficiente para lograr entrenar la red con buenos resultados.

Los archivos MIDI de esta base de datos poseen diferentes Tracks de varios instrumentos de los cuales en este proyecto solamente se tomaran en cuenta los Tracks de

guitarra eléctrica limpia, bajo eléctrico con dedos y batería, los demas Tracks serán ignorados.

3.1.2. Descripción del proyecto

Red Neuronal

La intención de este proyecto es crear una serie de redes neuronales que sean capaces de procesar las entradas de guitarra, bajo y batería, Para esto se usaron 3 redes neuronales por separado para procesar los instrumentos, estas redes tienen la siguiente arquitectura:

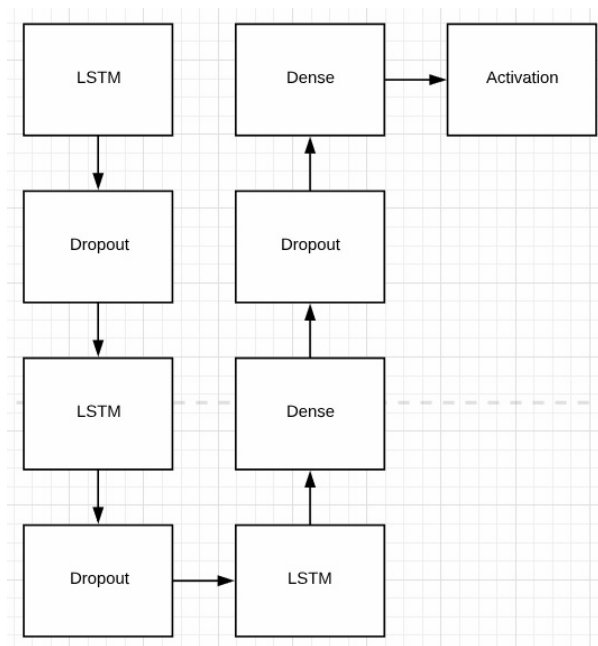


Figura 3.1: Estructura de la Red neuronal

Las capas de redes LSTM procesan de manera secuencial la información de las matrices generadas a partir de los eventos en los Tracks, las capas de Dropout son usadas para prevenir el sobre entrenamiento de la red, eliminando algunos datos de salida de las redes LSTM, finalmente llega a una capa Dense la cual nos ayuda al aprendizaje de la red.

La función de activación que se uso en esta red es una *tanh*, este tipo de activación permite de una manera mas suavizada la activación o desactivación de las neuronas en comparación con una señal sigmoidea

Arquitectura de la aplicación

La arquitectura de la aplicación se muestra en la siguiente figura:

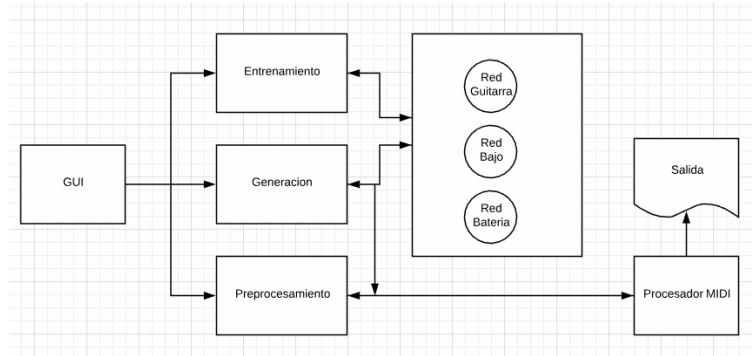


Figura 3.2: Arquitectura de la aplicación

En esta arquitectura se puede observar que se tiene varios módulos el sistema:

- 1.- **GUI.-** Este modulo contiene la interfaz de usuario para el programa.
- 2.- **Preprocesamiento.-** Este modulo se encarga de preprocesar los archivos MIDI con el fin de separar los tracks de guitarra, bajo y batería en diferentes archivos MIDI.
- 3.- **Entrenamiento.-** Este modulo se encarga de entrenar la red con los archivos MIDI previamente preprocesados.
- 4.- **Generación.-** Este modulo genera nuevas piezas a partir de enviar notas al azar a la red entrenada.
- 5.- **Redes neuronales.-** En este modulo se tienen 3 redes neuronales completamente independientes, una para guitarra, otra para bajo y la ultima para batería.
- 6.- **Procesador MIDI.-** Este modulo contiene todas las utilerías para el manejo de los archivos MIDI.

3.1.3. Proceso de entrenamiento

3.1.4. Validación de resultados

Capítulo 4

Conclusiones

Bibliografía

- [1] C. Raffel, “Learning-based methods for comparing sequences, with applications to audio-to-midi alignment and matching,” Ph.D. dissertation, Columbia University, 2016.
- [2] M. H. Hassoun, *Fundamentals of artificial neural networks*, 1995.
- [3] K. Gurney, *An Introduction to Neural Networks*, C. Press, Ed., 2003.
- [4] I. Nunes, D. Hernane, R. Andrade, L. Bartocci, and S. dos Reis, *Artificial Neural Networks: A Practical Course*, Springer, Ed., 2017.
- [5] N. Buduma, *Fundamentals of Deep Learning: Designing Next-Generation Machine Intelligence Algorithms*, O. Reilly, Ed., 2017.
- [6] L. Deng and D. Yu, *Deep Learning: Methods and Applications*, Foundations and T. in Signal Processing, Eds. now Publishers Inc, 2013.
- [7] J. Briot, G. Hadjeres, and F. Pachet, *Deep Learning Techniques for Music Generation - A Survey*, 2017.
- [8] E. Herrera, *Teoría musical y armonía moderna*, A. Bosh, Ed., 1984.
- [9] E. Taylor, *A.B. Guide to Music Theory*, ABRSM, Ed., 1989.
- [10] H. Lee, P. Pham, Y. Largman, and A. Y. Ng, “Unsupervised feature learning for audio classification using convolutional deep belief networks,” *MTT-S International Microwave Symposium Digest*, pp. 1096–1104, 2009.
- [11] E. J. Humphrey, J. P. Bello, and Y. LeCun, “Moving beyond feature design: deep architectures and automatic feature learning in music informatics,” *13th*

- International Society for Music Information Retrieval Conference*, pp. 403–408, 2012.
- [12] A. Pikrakis, “A deep learning approach to rhythm modelling with applications,” 2013.
- [13] X. Wang and Y. Wang, “Improving content-based and hybrid music recommendation using deep learning,” pp. 627–636, 2014.
- [14] S. Dieleman and B. Schrauwen, “End-to-end learning for music audio,” 2014.
- [15] A. Huang and R. Wu, “Deep learning for music,” 2016.