

# Clasificación de acordes con redes neuronales

Rodrigo Russo<sup>1</sup> y Andrés Oliva<sup>1</sup>

Dra. Ing. Lucía Isabel Passoni<sup>2</sup>, CC Ana Lucía Dai Pra<sup>2</sup>, Dr. Ing. Gustavo Meschino<sup>2</sup>

<sup>1</sup> Facultad de Ingeniería, Universidad Nacional de Mar del Plata, Laboratorio de Bioingeniería, Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICyTE) CONICET-UNMDP

<sup>2</sup> Cátedra de Modelización y Predicción con Tecnologías Emergentes  
Facultad de Ingeniería, Universidad Nacional de Mar del Plata, Laboratorio de Bioingeniería, Instituto de Investigaciones Científicas y Tecnológicas en Electrónica (ICyTE) CONICET-UNMDP

**Resumen** El presente trabajo propone el uso de redes neuronales supervisadas y no supervisadas para el reconocimiento de acordes musicales. Partiendo de una porción de un archivo de audio, se espera hallar cuál es el acorde que corresponde al mismo. A partir de un preprocesamiento del mismo, utilizando transformada de Fourier y Pitch Class Profile, se generan los conjuntos de datos a clasificar, siendo críticos en la clasificación los errores por ambigüedades y ruido ambiental que alteran la interpretación de los datos. Los métodos empleados consisten en clasificar a partir de grupos conocidos y bien definidos, utilizando redes neuronales supervisadas mientras, o bien, realizando un agrupamiento de los datos según características similares mediante mapas autoorganizados. Se espera que ambos métodos logren agrupar los datos en las categorías correspondientes y minimizar, en lo posible, los errores de clasificación.

## 1. Introducción

En teoría musical, un acorde corresponde a tres o más notas musicales sonando simultáneamente. Éstos pueden clasificarse según la distancia que haya entre las notas que lo componen.

Por lo general, en la mayor parte de la música, las combinaciones posibles pueden realizarse con doce notas diferentes siguiendo la siguiente convención:

Do, Do<sup>#</sup>/Re<sup>b</sup>, Re, Re<sup>#</sup>/Mi<sup>b</sup>, Mi, Fa, Fa<sup>#</sup>/Sol<sup>b</sup>, Sol, Sol<sup>#</sup>/La<sup>b</sup>, La, La<sup>#</sup>/Si<sup>b</sup> y Si

Cada una está separada de otra por lo que se conoce como un semi-tono, siendo un tono la distancia entre dos semitonos. Por otro lado, existe una forma específica de nombrar la distancia entre dos notas cualesquiera, denominada *intervalo*. A continuación se enumeran los distintos intervalos posibles, separados un semi-tono cada uno del siguiente:

## 2 Introducción

Unísono (fundamental), segunda menor, segunda mayor, tercera menor, tercera mayor, cuarta justa, cuarta aumentada/quinta disminuida, quinta justa, sexta menor, sexta mayor, séptima menor, séptima mayor, octava (fundamental)

Esta clasificación de intervalos es válida para distancias menores que una octava (que será la fundamental al doble de la frecuencia). Dado que las notas pueden repetirse más allá de la octava, en nuestro caso, esta clasificación se repetirá siendo la octava equivalente al unísono. De esta forma una tercera mayor por encima de una octava, será considerada una tercera mayor.

Recordando la definición de acorde, se tienen tres o más notas en simultáneo separadas entre sí de ciertos intervalos. Es necesario definir sobre qué nota corresponde el intervalo pues entre Do y Sol existe un intervalo, pero entre Sol y Do existe otro. Entonces se tomarán todos los intervalos respecto a una nota particular del acorde denominada *fundamental* la cual será la que dote del nombre a éste. Por ejemplo, en un acorde de Do mayor, su fundamental será Do.

Dado que tomar todas las combinaciones excede el interés de este trabajo, se tendrán en cuenta las principales y comúnmente utilizadas. Estas combinaciones corresponden a los acordes mayores y menores, en total veinticuatro, siendo dos por cada nota. Un acorde mayor o menor está compuesto de tres notas: la fundamental, una tercera y la quinta. Cual sea la tercera es lo que define si es mayor o menor.

Las distancias correspondientes desde la fundamental serían tres semitonos para una tercera menor, cuatro semitonos para una tercera mayor y siete semitonos para la quinta justa. De esta forma, si el acorde en cuestión fuese Do mayor, estará compuesto de las siguientes notas: Do, Mi, Sol. Otro ejemplo podría ser Si<sup>b</sup> menor, compuesto por: Si<sup>b</sup>, Re<sup>b</sup>, Fa.

### 1.1. Cuantificación del problema

Dado que la mayor parte de la música puede hallarse digitalmente, por lo que es conocida la serie temporal que contiene la información del acorde a analizar; los tres tonos que lo componen. El caso real ( una grabación musical) resulta complejo de analizar teóricamente, por lo tanto, se partirá de un caso ideal más básico para obtener una forma de *medir* el acorde.

Una nota sera representada como una onda senoidal a cierta frecuencia específica  $f$  como  $x = \text{sen}(2\pi ft)$ . Se puede aprovechar la correspondencia que hay entre frecuencia y notas musicales. La distancia entre una nota y otra es logarítmica de base 2 y existen tablas que dan las frecuencias de las doce notas dentro de una octava. Para extenderlo a otras octavas, simplemente, se multiplica cada valor por una potencia de dos (de aquí por qué la base del logaritmo es 2). Sin embargo, como se busca obtener un acorde, la expresión de la señal debe ser una superposición de senoidales.

$$x(t) = \text{sen}(2\pi f_1 t) + \text{sen}(2\pi f_2 t) + \text{sen}(2\pi f_3 t) \quad (1)$$

Al mismo tiempo, esto trae otro tipo de problemas, como por ejemplo la afinación (que no necesariamente es siempre igual) y la altura de las notas porque

no siempre están dentro de una misma octava. La solución elegida consiste en normalizar todas las frecuencias a una octava y definir doce rangos de valores que corresponden a cada una de las notas.

Como se utilizarán grabaciones digitales, no es necesario que todas tengan la misma cantidad de muestras, sino que termina reduciéndose a doce variable, una por cada nota, lo cual es denominado *Pitch Class Profile* (PCP [1999]) que corresponde a una medida de la potencia de cada nota a lo largo de todo el rango audible. Conceptualmente, toma todas las deltas (ahora deltas de Kroenecker debido a que se trabaja con muestras) del plano frecuencial y las agrupa por nota. Si pertenece al rango donde se encuentra Do, entonces se asume que es Do y se suman acumulativamente todas las potencias que estén en ese rango.

### 1.2. Adquisición de los acordes

A lo largo de este trabajo se utilizó la herramienta MATLAB® que cuenta con una función para obtener las muestras de un archivo `.wav`. Esto diferirá del caso ideal ya que contendrá ruido, armónicos e información adicional que hace que un instrumento musical no suene como una onda senoidal pura. En principio, se analizarán archivos de audio que contengan un solo acorde (sea un grabación particular o una porción de una pieza musical).

### 1.3. Solución propuesta

Se busca que estos patrones (conjunto de acordes) mencionados anteriormente puedan ser clasificados en los veinticuatro acordes mayores y menores posibles, por lo tanto se proponen dos formas de realizarlo: la primera utilizando redes neuronales supervisadas y, la segunda, utilizando mapas autoorganizados.

## 2. Metodología

### 2.1. Red neuronal supervisada

El tipo de red neuronal a utilizar corresponde a una red multicapa hacia adelante (*feed-forward*) con una capa oculta de cantidad de neuronas a determinar. Como corresponde a una clasificación, la salida serán neuronas de función sigmoidea. Respecto a las entradas, consiste en un vector de doce elementos correspondiente al PCP de la muestra que contiene la potencia normalizada de cada nota. La salida corresponde a un vector de veinticuatro elementos que indica cuánto se asemeja a cada categoría.

**Entrenamiento:** Los conjuntos de entrenamiento consistirán en PCPs de muestras de entrenamiento. Dado que es posible obtenerlo a partir de una señal de audio, puede sintetizarse a partir de ondas senoidales (espectro ideal) y cuadradas (dispersión espectral) u obtener muestras reales. La elección de los conjuntos

#### 4 Metodología

de entrenamiento resulta crucial en este problema pues las muestras que se ingresarán son completamente desconocidas debido a diferentes factores: equipamiento de grabación, instrumento o ruido. A la vez, también es necesario contar con una cantidad de muestras para entrenar que genere el mínimo error de entrenamiento y a la vez realice una buena generalización. Por lo tanto, se ha procedido a experimentar con los siguientes conjuntos:

- Ondas senoidales ideales
- Ondas senoidales y cuadradas ideales
- Ondas senoidales y cuadradas ideales con y sin ruido aleatorio
- Ondas senoidales y cuadradas ideales con y sin ruido aleatorio con potencia máxima aleatoria
- Muestras reales

De todos estos conjuntos, se analizarán los primeros tres. Se descarta el cuarto ya que no produce resultados especialmente interesantes. Además, al existir un caso ambiguo, la red neuronal será capaz de clasificarlo un tanto por ciento en una categoría y otro tanto por ciento en otra. Respecto a la quinta, se requiere de una diversa cantidad de instrumentos y condiciones para obtener variedad. Dada la naturaleza de los PCP, se intentará obtener buenos resultados únicamente utilizando muestras sintetizadas. El procedimiento consiste en decidir un número de muestras (veinte, en principio) por cada uno de los veinticuatro acordes y generarlos según el caso. En el primero, simplemente serán ondas senoidales cuyas componentes espectrales corresponden a los tonos que forman el acorde. La segunda alterna entre ondas senoidales y cuadradas de forma aleatoria. En estos dos casos habrán redundancias, por lo tanto se analizará si afecta al entrenamiento. El último conjunto es igual al anterior con la diferencia que la mitad de las muestras están sometidas a ruido aleatorio.

El método que se utilizó para generar este entrenamiento es un algoritmo que genera determinada cantidad de acordes de cada tipo; si se desean cinco acordes en cada tonalidad, se esperan entrenar un total de 120, recordando que son veinticuatro tonalidades distintas.

El objetivo del entrenamiento es que pueda detenerse logrado un buen desempeño y minimizando el gradiente, de esta forma garantizando un error mínimo.

De cada tipo de entrenamiento se realizaron cuarenta de cada uno y se promediaron los errores. En primera instancia, esto se realizó para conocer si algún conjunto de entrenamiento tiene la capacidad de ser entrenado mejor. En promedio, el primero tuvo el 51 % de error de entrenamiento, el segundo 50 % y el tercero 50 %. La proximidad de estos valores indican que ningún conjunto tiene alguna característica en particular que logre mejores resultados. Sin embargo, es necesario encontrar cuáles de ellos obtuvieron el 0 % de error. Como todas las muestras son *ideales*, lo mínimo que se espera de la red es que pueda clasificarlas a la perfección. Si no es posible clasificar un acorde ideal, menos se puede esperar de las muestras reales. Para realizarlo, se formó un conjunto de entrenamiento y se entrenaron cientos de redes neuronales almacenando, únicamente, las que no producían error. Al mismo tiempo, se fueron variando el número de neuronas

en un rango de 50 a 120 aproximadamente. Las que, según el criterio anterior, generaron *mejores* redes produjeron las de 70 y 80 neuronas; por lo tanto, se tomó como número de neuronas a 75 en la capa oculta. Cabe destacar que la repetición de los patrones produjo mejores resultados en el entrenamiento.

Si bien los conjuntos de entrenamiento se mostraron útiles a la hora de obtener redes de cero error, todos los conjuntos lograron entrenar satisfactoriamente múltiples redes. Con esto se observa que ninguno resulta más mucho mejor que otro respecto al entrenamiento. Sin embargo, lo importante es la capacidad de generalización de cada red. En la sección siguiente, se probará cada red con diferentes tipos de muestras reales.

**Generalización:** En esta sección se comprobará el comportamiento de cada red neuronal generada por cada uno de los conjuntos de entrenamiento. Las redes cuyo error fue mayor al 0% son descartadas. Los conjuntos que se evaluarán en la red neuronal serán acordes en piano, guitarra y cuarteto de cuerdas sintetizados en un programa secuenciador de MIDI formando diferentes patrones y acordes de la canción previamente mencionada *Komm süßer, tod*. El primero de los conjuntos resulta ser el más ideal”, pues utiliza muestras grabadas en un ambiente de poco ruido; el segundo consta del caso deseado a analizar.

Antes de proceder a los resultados, se mostrarán cada patrón usados en las muestras ideales. Estos se repiten y trasponen para cada uno de los acordes, entonces, mostrar sólo el de Do mayor, basta. Lo que se busca hallar es si la red es capaz de reconocer un acorde en varios instrumentos por separado y a la vez. Las muestras tomadas a partir de la canción, no es necesario mostrarlas pues no necesariamente responden a un patrón en particular.

El conjunto de entrenamiento más complejo, resultó tener en promedio mejores resultados. Dado que contempla casos más aleatorios y reales, se esperaba que se comporte mejor que los demás con muestras reales.

Siendo los resultados obtenidos no muy diferentes al utilizar los diferentes conjuntos de entrenamiento. A partir de la experimentación, varias redes entrenadas con los mismos conjuntos de entrenamiento, no suelen dar los mismo resultados; es decir, dos redes pueden clasificar de distinta forma (aunque similarmente) un acorde. Intuitivamente, al entrenar una cantidad importante de redes que cumplan las condiciones dichas, la mayoría debe converger al acorde más *lógico* según el PCP obtenido. De esta forma, se emplearon dos métodos de cálculo de salida para verificar que esto se cumple. Uno consiste en promediar todas las salidas y definir un acorde y otro consiste en tomar el que se repite la mayor cantidad de veces.

**Promedios:** Para verificar que promediando salidas se mejora el resultado, se entrenaron grandes cantidades de redes neuronales. En la siguiente tabla, se muestran las muestras usadas para verificar la capacidad de generalización de las redes pero aplicando este principio para 3, 7 y 56 redes:

Siendo el resultado en una mejora notable al emplear una exageradamente mayor cantidad de redes. Lo mismo podría suceder con las muestras reales, por lo tanto, se procederá a observar los resultados para el mejor caso (56 redes):

## 6 Metodología

Donde se mantuvo en este caso el error del 44,4 % y con los errores en los mismos acordes. Por lo tanto, se procede al caso del valor más repetido.

**Valor más repetido:** Análogamente al caso anterior, se presenta el mismo cuadro con el nuevo método pero obviando el caso de 3 redes.

|                     | Sin promedios | Valor más repetido<br>(7 redes) | Valor más repetido<br>(56 redes) |
|---------------------|---------------|---------------------------------|----------------------------------|
| Piano 1             | 3 (12,5 %)    | 2 (8,33 %)                      | 1 (4,16 %)                       |
| Piano 2             | 10 (41,66 %)  | 8 (33,33 %)                     | 6 (25 %)                         |
| Piano 3             | 4 (16,66 %)   | 7 (29,17 %)                     | 3 (12,5 %)                       |
| Guitarra 1          | 10 (41,66 %)  | 11 (45,83 %)                    | 9 (37,50 %)                      |
| Guitarra 2          | 12 (50 %)     | 9 (37,50 %)                     | 10 (41,66 %)                     |
| Cuarteto de cuerdas | 8 (33,33 %)   | 9 (37,50 %)                     | 7 (29,17 %)                      |
| Promedio de errores | 32,63 %       | 31,94 %                         | 25 %                             |

**Cuadro 1.** Resultados de las muestras de una canción

Finalmente, queda observar si mejora la clasificación de un caso real con este nuevo método de clasificación:

| Acorde esperado       | Acorde obtenido             |
|-----------------------|-----------------------------|
| Si <sup>b</sup> mayor | <b>Si<sup>b</sup> mayor</b> |
| Re menor              | <b>Re menor</b>             |
| Fa mayor              | <b>Fa mayor</b>             |
| Sol menor             | <b>Sol menor</b>            |
| Re menor              | <b>Re menor</b>             |
| Mi <sup>b</sup> mayor | Mi <sup>b</sup> menor       |
| Si <sup>b</sup> mayor | <b>Si mayor</b>             |
| Mi <sup>b</sup> mayor | Mi <sup>b</sup> menor       |
| Fa mayor              | Fa menor                    |
| Errores               | <b>3 (33,3 %)</b>           |

Presenta una leve mejora en este caso, sin embargo, hay que notar que, la mayoría de las veces, los últimos dos acordes no fueron correctamente clasificados. Eso puede deberse a un problema de ambigüedad y la clasificación sea correcta desde el punto de vista del modelo planteado.

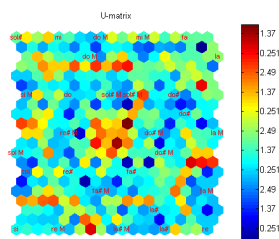
## 2.2. Mapas autoorganizados

El vector de entrada consiste en la potencia normalizada (de valor entre 0 y 1) de cada una de las 12 notas musicales que componen un acorde, siendo en este caso 9 aproximadamente iguales a 0, y 3 con un valor más alto próximo a 1, puesto que un acorde se compone por 3 notas.

La herramienta utilizada fue MATLAB (algoritmos de SOM de Kohonen).

Se utilizó la función generadora de acordes mencionada previamente para poder realizar un entrenamiento, partiendo de la hipótesis del uso de datos ideales (sin ruido) , al suponerse que sería más fácil realizar la separación entre clusteres por parte de la red.

Se tomó el número arbitrario de generar 10 acordes de cada tipo (obteniéndose 240 vectores de entrada) para realizar el primer entrenamiento , utilizando los parámetros por defecto en los demos.



**Figura 1.** U- Matriz resultante de la primer red neuronal.

Se pudo observar que la separación de los clusters de la red (y por ende, la capacidad de la red para "diferir" entre cada cluster) era bastante difusa.

Como primer solución, se procedió a determinar con que parametros de entrenamiento (forma, función vecindad, ect) se obtendrían mejores resultados, desarrollando un algoritmo que genera redes a partir de todas las combinaciones posibles, para luego elegir la combinación cuya característica fuera tener el error topográfico más bajo y menor error de cuantificación posible como prioridad. Se partió de la hipótesis (al observar en reiterados casos casos), que por el criterio de evaluación utilizado en esta etapa, la separación de los clusteres era más "marcada" cuando el error topográfico era menor.

La configuración que menor error dio fue una red de tamaño "grande", cuadrícula "Hexagonal", forma "Sheet", función vecindad "Gaussiana", entrenamiento "Largo", inicialización "Aleatoria" y entrenamiento por "Lotes".

A partir de esta selección, debido a que la configuración elegida presenta una inicialización aleatoria, se utilizó un algoritmo que consistía en generar un número N de redes (1000 a 10000 según el número de acordes), para luego seleccionar aquella que presentara el menor error, priorizando el topográfico.

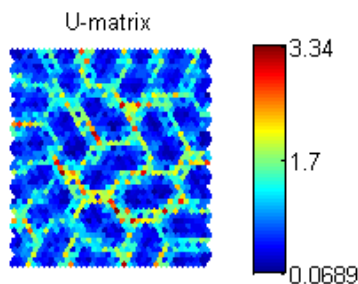
Al haber obtenido una mejora respecto del primer caso, se procedió a analizar el objetivo del trabajo: la formación de clusteres y su uso para la clasificación.

Para realizar la clasificación, se utilizó el algoritmo **kmeans** (por lotes, al ser el mismo utilizado en el entrenamiento del mapa) siendo el máximo número de clusteres generados igual a 24, puesto que el número de posibles acordes.

Se obtuvo que el error es más bajo para el caso en el que se realizan 24 clusters. Pruebas posteriores utilizando un N mayor (40) muestran que el error

## 8 Metodología

sigue decreciendo hasta 27 clusters, para luego aumentar enormemente, lo que se puede interpretar como una cierta tendencia natural en los datos a ser agrupados de esta manera.



**Figura 2.** U- Matriz resultante al utilizar los parámetros óptimos de entramiento.

La cantidad de clusteres generados fue de 1000 en este caso, eligiendo aquel cuyo error fuera el menor como óptimo. Se tomaron secciones del algoritmo de kmeans, para que sólo generara clusteres de 24, al ser este el número buscado.

Una vez obtenido el cluster óptimo respecto del error, surgió la incógnita de cómo llevar a cavo el proceso de clasificación, y más importante, cómo determinar qué acorde corresponde a qué cluster.

Para determinar a qué cluster pertenece cada entrada, se utilizó la distancia euclideana, al ser la que se utilizó en el entrenamiento del mapa, con la que se obtuvo por amplia diferencia los mejores resultados experimentales. Sin embargo, entra en juego nuevamente la pregunta anterior: *¿Cómo determino que cluster representa cada posible acorde?*

Para solucionar el problema, se utilizaron los datos de entrenamiento..

Partiendo de la hipótesis de que si los clusters fueron formados apropiadamente, entonces al ingresar los datos de entrada (que al ser generados, se conoce unívocamente que acorde corresponde) deberían ser clasificados perfectamente.

Es decir, siguiendo el ejemplo inicial, al tener 10 datos para cada acorde, luego de realizar la clasificación, si el sistema funcionase perfectamente, debería de tener 10 acordes asociados a cada cluster.

De esta hipótesis se creo un algoritmo que realizara esta asociación, consistiendo en tomar todas las entradas pertenecientes al mismo acorde, determinado a que cluster son asignadas. Si todas las entradas que corresponden al mismo cluster, son asignadas al mismo, entonces se asigna a susodicho cluster la etiqueta al acorde correspondiente (Por Ejemplo, todos los acordes que deberían ser Do mayor son asignados al cluster 7, entonces se etiqueta el cluster 7 como Do mayor). Basta con que un acorde sea asignado incorrectamente para que el sistema identifique ese cluster como no valido, asignado la etiqueta cero.



Posteriormente se comprobó que es determinante que el sistema pueda clasificar apropiadamente los datos de entrenamiento para poder reconocer correctamente acordes distintos de los mismos (externos).

Este método se utilizó para poder realizar una evaluación más apropiada y consistente a la calidad de la red; al asignar las etiquetas, se realiza una clasificación de los datos de entrenamiento, y se compara el resultado con el valor verdadero unívocamente, obteniéndose un error porcentual a partir de comparar los resultados obtenidos con los esperados.

Para la primera red, mencionada al principio, el resultado fue de un error del 95,5%, con varios casos en donde impactaban hasta 5 o 6 acordes diferentes en un mismo cluster. En el caso de la segunda red, el error fue del 88%, con la consideración de que la cantidad de acordes distintos por cluster fue de un máximo de 2, lo cual fue una clara mejora en el sistema.

Se realizaron pruebas con datos externos, pero los resultados fueron igualmente pobres. Ya habiendo adquirido un método para realizar una evaluación más "funcional" del sistema, en el sentido que se basa directamente en la capacidad de clasificación del mapa en lugar de coeficientes matemáticos de error, se empleó este algoritmo en forma integrada con el encargado de generar los clusters, de modo de obtener el cluster con el menor error de clasificación posible.

Es de destacar que el hecho de que sea posible realizar los agrupamientos claros no significa exactamente que no haya error al clasificar los datos de entrenamiento: Por ejemplo, a los clusters 15 y 21 se les asignan la misma etiqueta 2, por lo tanto, si el error de generación de claves fue de 0%, entonces habrá 2 clusters con la misma etiqueta, resultando en una etiqueta con un dato

Asimismo, se implementó un nuevo algoritmo para generar mapas de distinta cantidad de grupos de acordes de entrada. Cabe aclarar que para los casos de mayor cantidad de datos de entrada (480; es decir, 20 vectores por acorde) se debió aumentar la cantidad de iteraciones para obtener resultados comparables respecto del error.

El mejor resultado fue, irónicamente, el utilizado para entrenar: siendo 240 datos de entrada (10 por acorde) y 216 (9 por acorde). Lográndose un 0% de error de etiquetas, y un error de clasificación del 4,066% para ambos casos. Entre 5 y 12 datos de entrenamiento, el error se mantuvo en un 4-8%, mientras que en los demás casos se disparó hasta un 20%.

También se experimentó entrenar un mapa utilizando datos de entrada no ideales (con ondas cuadradas, y ruido, respectivamente).

Finalmente, se sometió a prueba con datos externos, para comprobar la fiabilidad del sistema. En el siguiente cuadro, se comparan los porcentajes de error para dichos casos:

## 10 Conclusiones

| Tipo de muestras          | 240     | 216            |
|---------------------------|---------|----------------|
| Piano 1                   | 4,16 %  | 4,16 %         |
| Piano 2                   | 20,83 % | 10,41 %        |
| Piano 3                   | 10,41 % | 6,25 %         |
| Guitarra 1                | 12,50 % | 12,50 %        |
| Guitarra 2                | 25 %    | 20,83 %        |
| Komm Susser Tod           | 33,33 % | 33,33 %        |
| Wings of Destiny          | 11,11 % | 22,22 %        |
| Porcentaje total de error | 16,75 % | <b>15,67 %</b> |

Por lo que se decidió tomar el caso de 216 como óptimo. Es interesante destacar que el mapa no funciona apropiadamente a la hora de utilizar datos con ruido como entrada. Se supone que este comportamiento es debido al hecho de que el ruido introducido en los vectores que deberían ser nulos altera las distancias entre los elementos y al ser la distancia euclidiana un valor numérico (modulo) que representa una distancia de 12 variables, no tiene noción de cuales son los elementos del vector mas o menos distantes, sino de la distancia total del arreglo.

### 3. Conclusiones

**Red Neuronal Supervisada:** Finalmente, tras más ensayos, se puede observar una clasificación *en general* correcta. Logra captar la idea general de los acordes que participan de una porción de audio pero no todos. Especialmente sufre de complicaciones a la hora de agregar una mayor cantidad de instrumentos (que a la vez agrega mayor contenido espectral). Posibles formas de solucionar esto es obtener un algoritmo mejor de PCP que realice alguna especie de filtrado o ventaneo en particular.

Por otro lado, existe el problema del entrenamiento. Para ser entrenado con casos completamente ideales, tiene un comportamiento mejor que el esperado. Aún así, se espera que con muestras reales de calidad se puedan mejorar los resultados para casos particulares. Por ejemplo, si se quisiera analizar guitarras, entrenar con guitarras y así sucesivamente.

**Mapas Autoganizados:** Durante la realización de este trabajo se logro desarrollar un método para atacar una problemática en donde inicialmente la cantidad de variantes respecto al entrenamiento y clasificación de los datos, era bastante alta, lo cual dificultaba la proyección del proyecto respecto de la dirección a tomar para el cumplimiento de los objetivos.

Se logro aprovechar la característica de los mapa autoorganizados a la hora de mostrar la información, para la formulación y chequeo de hipótesis respecto de los resultados esperados a la hora de realizar la clasificación.

Es de destacar que los métodos detallados en este trabajo se aplicaron en forma iterativa.

Respecto al objetivo, si bien el método no funciona en forma ideal, se consideran satisfechos los resultados tomando en consideración el punto de partida del trabajo, y los avances que se lograron al crear metodologías de trabajo y evaluación concisas con el mismo.

**Metodología:** Teniendo en cuenta el funcionamiento y la capacidad de clasificación correcta presentada en los metodos, se puede decir que para clasificar perfectamente acordes no es muy fiable. Sin embargo, dado que se puede captar una noción general de la tonalidad de la música, puede servir para analizar una pieza en cuanto a percepción y para una rápida clasificación por tonalidad. Para esto último, puede utilizarse un algoritmo similar al de valor más repetido viendo en qué tonalidad pertenece la mayoría de los acordes.

## Referencias

- [1999] Fujishima, Takuya. Realtime Chord Recognition of Musical Sound: a System Using Common Lisp Music. Stanford University
- [2012] NEURAL NETWORKS FOR MUSICAL CHORDS RECOGNITION J. Osmalskyj, J-J. Embrechts, S. Piérard, M. Van Droogenbroeck INTELSIG Laboratory, University of Liège, Departement EECS