Desarrollo de un sistema de ecualización automática de audio a partir de un modelo de incrustación de palabras

Trabajo de grado

Ingeniería de Sistemas y Computación

Juan David Rengifo Mera - 8938889

3 de septiembre de 2024

Supervisor: Dr. Gerardo Sarria

Índice general

Índice de figuras

Capítulo 1

Introducción

La ecualización de audio es un proceso que consiste en la modificación del contenido de frecuencia a través de ganancias positivas o negativas, cambiando así las características armónicas y tímbricas del audio. Este proceso se utiliza para corregir problemas de frecuencia generados por acústica y dispositivos de captura, además se utiliza para optimizar grabaciones y realizar ajustes de mezcla que brinden equilibrio tonal [10].

Actualmente, los sistemas de ecualización de audio se basan principalmente en el conocimiento experto de los ingenieros de sonido y su experiencia en el uso de ecualizadores de audio. Sin embargo, este proceso puede ser lento y subjetivo, y a menudo requiere ajustes manuales para lograr el resultado deseado [11]. Además, los ingenieros de sonido tienen diferentes niveles de experiencia y preferencias personales, lo que puede conducir a resultados inconsistentes. Por lo tanto, se han propuesto diversas alternativas para hacer que los sistemas de ecualización de audio sean más precisos y automáticos, mejorando la eficiencia y la calidad del proceso de producción de sonido.

Existen varios métodos para automatizar el proceso de ecualización de audio, uno de ellos es la automatización basada en frecuencia. Este método utiliza algoritmos de procesamiento de señal para analizar la frecuencia y el contenido espectral del audio. Estos algoritmos pueden ajustar automáticament la ecualización del audio en función de la frecuencia y el espectro, logrando una mezcla de sonido más equilibrada y coherente. Por otro lado, está la automatización basada en modelos aprendizaje automático para analizar el contenido de audio y ajustar la ecualización automáticamente de acuerdo a características del sonido. Por útlimo, están los métodos basados en algoritmos de optimización que pueden encontrar el ajuste óptimo para la ecualización del audio en función de ciertos criterios, como el nivel de presencia de las frecuencias o la calidad de la mezcla de sonido. En general, cada uno de estos métodos tiene sus propias ventajas y desventajas y se pueden utilizar en diferentes contextos dependiendo de las necesidades específicas de la producción de sonido [10].

Particularmente, hay distintas formas de realizar ecualización automática con modelos de aprendizaje automático. Una aproximación interesante es el uso de modelos de incrustación de palabras. Esta es una técnica de procesamiento

del lenguaje natural que permite representar las palabras como vectores numéricos densos en un espacio vectorial de alta dimensión. Cada palabra se representa por un vector de números reales que captura el significado semántico y sintáctico de la palabra, así como sus relaciones con otras palabras. La idea detrás de estos modelos consiste en que las palabras que aparecen en contextos similares tienden a tener significados similares, por lo tanto, las palabras que tienen vectores similares se consideran que son semánticamente similares o tienen un significado similar.

En contexto, la incrustación de palabras aplicada a la ecualización automática implica utilizar descripciones de audio en forma de texto como entrada para ajustar automáticamente la respuesta de frecuencia del sistema de audio, utilizando algoritmos de procesamiento de lenguaje natural y modelos de aprendizaje automático para generar una respuesta de ecualización adecuada para cada descripción de audio [11].

Ahora bien, entrenar un modelo para generar ecualizaciones de audio necesita de ecualizaciones anotadas con descripciones humanas. Los autores Cartwright y Pardo [13] desarrollaron una solución para el problema de las anotaciones presentando un conjunto de datos llamado SocialEQ. Este proyecto se basa en una plataforma en línea que utiliza el crowdsourcing para recopilar información y aprender un vocabulario de descriptores de audio. Esto hace posible realizar ecualizaciones automáticas de audio utilizando descriptores semánticos en inglés (e.g. "*make it warmer*").

1.1. Planteamiento del problema

Conseguir un sonido uniforme y equilibrado en la producción de audio es un reto habitual que afecta a expertos en una gran variedad de campos, como la producción musical, la creación de podcasts, la industria del cine y la televisión y el desarrollo de videojuegos. Una de las principales razones de este reto radica en que las grabaciones de audio pueden llevarse a cabo en diferentes entornos, con distintos tipos de equipamiento y con diferentes niveles de experiencia.

Por ejemplo, un productor de música puede grabar las voces en un estudio profesional con micrófonos de alta calidad, mientras que graba pistas de guitarra en un estudio casero con un micrófono de menor calidad. Estas dos grabaciones pueden tener respuestas de frecuencia distintas, lo que dificulta lograr un sonido uniforme y equilibrado en ambas pistas. Asimismo, un productor de podcasts puede grabar entrevistas en una gran variedad de ubicaciones, cada una con características acústicas únicas, resultando en grabaciones con niveles de claridad y equilibrio diferentes. Aunque la ecualización de audio manual es una técnica comúnmente utilizada para abordar estos problemas, puede ser un proceso subjetivo y que requiere mucho tiempo, experiencia y conocimientos. Incluso expertos en audio experimentados pueden tener dificultades para lograr resultados consistentes en diferentes grabaciones.

Adicionalmente, la ecualización además de solucionar problemas de frecuencia, se presta a una gama de posibilidades creativas que permiten a los pro-

ductores y mezcladores de audio dotar de un matiz propio y distintivo a las mezclas. Al implementar la ecualización de forma creativa, es factible obtener efectos emocionales específicos.

Un ejemplo común es el aumento de las frecuencias altas para que un sonido se perciba más brillante y emocionante, o la reducción de las frecuencias graves para que suene más distante y sutil. No obstante, dado que la terminología asociada a la ecualización es técnica y especializada, los músicos y artistas podrían tener problemas para comunicar sus ideas a los ingenieros de sonido y productores. Como resultado, los productores deben entender el vocabulario semántico utilizado por los músicos y artistas para aplicar la ecualización de manera apropiada y lograr el sonido deseado, lo que puede ser un desafío adicional.

Por consiguiente, en el presente trabajo, se pretende especificar una herramienta que utilice modelos de aprendizaje profundo para realizar ecualizaciones automáticas de audio basadas en descripciones en lenguaje natural. Esto con el fin de hacer accesible para cualquiera ecualizaciones de audio que respondan a sus deseos creativos.

1.2. Formulación

¿De qué forma se puede implementar una herramienta de ecualización automática basada en descripciones del sonido haciendo uso de modelos de aprendizaje profundo y procesamiento del lenguaje natural?

1.3. Sistematización

Se busca dar respuesta a las siguientes preguntas:

- ¿Cómo recolectar una gran cantidad de datos que agrupen grabaciones de audio con sus respectivas descripciones y características del sonido?
- ¿Cómo construir un modelo de palabras incrustadas para representar las descripciones semánticas de los ajustes de ecualización como vectores numéricos?
- ¿Cómo construir un modelo de aprendizaje profundo para entrenar en el conjunto de datos de audio etiquetado reconociendo patrones en los datos de audio que corresponden a los ajustes de ecualización utilizados en la grabación original?
- ¿Cómo combinar los modelos de aprendizaje profundo y palabras incrustadas para crear un modelo que pueda tomar como entrada una descripción semántica de un ajuste de ecualización y producir como salida un conjunto de ajustes de ecualización para lograr el sonido deseado?
- ¿Cómo integrar el modelo en un sistema de ecualización automático que pueda procesar archivos de audio en tiempo real?

1.4. Objetivos

1.4.1. Objetivo General

Desarrollar una herramienta de ecualización automática basada en descripciones del sonido haciendo uso de modelos de aprendizaje profundo y procesamiento del lenguaje natural.

Objetivos específicos

- **O1:** Recopilar un conjunto de datos de audio etiquetado con información de ecualización.
- **O2:** Utilizar un modelo de aprendizaje profundo para entrenar en el conjunto de datos de audio etiquetado.
- **O3:** Utilizar un modelo de palabras incrustadas para representar las descripciones semánticas de los ajustes de ecualización como vectores numéricos.
- **O4:** Predecir el valor de ecualización de 40 variables a partir de una palabra que describa semánticamente un sonido.
- **O5:** Desarrollar y validar un sistema de ecualización automático basado en el modelo entrenado que pueda procesar archivos de audio y aplicar los ajustes de ecualización sugeridos.
- **O6:** Realizar una serie de pruebas con usuarios para evaluar la eficacia y usabilidad del sistema de ecualización automático, recogiendo feedback para posibles mejoras y refinamientos.

1.5. Delimitaciones y Alcances

Considerando que el desarrollo del proyecto depende de modelos de aprendizaje profundo, el alcance del mismo está limitado por la disponibilidad y calidad de los datos de entrenamiento, la disponibilidad de recursos computacionales capaces de realizar las tareas, la complejidad del modelo, la precisión con que los modelos pueden estimar valores y su capacidad para ser implementado en tiempo real.

De esta forma se quiere incluir la evaluación del rendimiento de los modelos utilizados mediante pruebas comparativas con otros métodos de ecualización y la medición de la calidad de sonido resultante, incluir el análisis de la eficiencia computacional de los modelos y la identificación de posibles mejoras para optimizar el uso de recursos y comparar el desempeño de los modelos de aprendizaje profundo con otros enfoques de ecualización, como la ecualización manual (hecha por algún experto) o el uso de otros métodos de procesamiento de señales de audio.

Capítulo 2

Estado del arte

2.1. Áreas Temáticas

A continuación, se presentan las categorías relacionadas con este proyecto:

- Electrónica, Ciencias de la computación, Matemática aplicada → Procesamiento de señales → Procesamiento de señales de audio.
- Ciencias de la computación → Inteligencia artificial → Aprendizaje automático → Aprendizaje profundo.
- Ciencias de la computación, Matemática → Inteligencia artificial, estadística → Aprendizaje automático → Métricas de evaluación.

2.2. Trabajos relacionados

- Moffat *et al* [12]. En este trabajo se propuso una red neuronal profunda de basada en la arquitectura Wave-U-Net para realizar mezclas automáticas de batería. Se siguió un enfoque de extremo a extemo donde el audio sin procesar de las grabaciones de batería individuales es la entrada del sistema y la forma de onda de la mezcla estéreo es la salida. De esto se reporta que las mezclas generadas por el modelo propuesto son prácticamente indistinguibles de las mezclas humanas profesionales y, al mismo tiempo, superan los enfoques de mezcla inteligente anteriores.
- B. De Man *et al* [13]. En este trabajo se introdujo un motor de mezcla que hace uso de reglas de mezcla semánticas y basa las decisiones de mezcla en etiquetas de instrumentos, así como en características de señal elementales de bajo nivel. Las reglas de mezcla se derivan de los libros de texto prácticos de ingeniería de mezcla. El rendimiento del sistema se comparó con las herramientas de mezcla automática existentes, así como con ingenieros humanos mediante una prueba de escucha.
- Moffat *et al* [14]. Se propuso una representación utilizando el Rule Interchange Format (RIF) comúnmente utilizado en la Red Semántica. Los sistemas con diferentes capacidades pueden usar el razonamiento OWL en esos conjuntos de reglas de mezcla para determinar los subconjuntos

que pueden manejar adecuadamente. Se demostró esto por medio de una herramienta red de ejemplo que utiliza un solucionador de restricciones lógicas para aplicar las reglas en tiempo real a conjuntos de pistas de audio anotadas con características.

- Reiss *et al* [15]. Se desarrolló un dispositivo de mezcla de adaptación cruzada con el fin de optimizar los niveles de ganancia de una mezcla de audio en vivo. El objetivo del método es lograr niveles de mezcla óptimos mediante la optimización de las proporciones entre la sonoridad de cada canal de entrada individual y la sonoridad general contenida en una mezcla estéreo. Para evaluar la cantidad de sonoridad de cada canal en tiempo real, se realizaron mediciones estadísticas acumulativas. El sistema utiliza un algoritmo de adaptación cruzada para asignar los indicadores de volumen a los valores de ganancia del canal. El sistema tiene aplicaciones en mezcla automática de música en vivo, mezcla en vivo de audio de juegos y posproducción de grabación en estudio.

- E. T. Chourdakis *et al* [16]. Se propuso un diseño de un efecto de audio digital adaptativo para reverberación artificial, controlado directamente por las características de reverberación deseadas, que le permite aprender del usuario de forma supervisada. El usuario proporciona ejemplos monofónicos de las características de reverberación deseadas para pistas individuales tomadas del Open Multitrack Testbed. Se usó estos datos para entrenar un conjunto de modelos para aplicar automáticamente la reverberación a pistas similares. Por último, se evaluaron esos modelos usando clasificadores f1-scores, errores cuadráticos medios, y pruebas de escucha multiestímulo.

- M. Cartwright *et al* [17]. En este trabajo, se especifica SocialEQ, un proyecto basado en la web para aprender un vocabulario de audio accionable basado en descripciones de ecualizaciones. Desde su implementación, SocialEQ ha aprendido 324 palabras distintas en 731 sesiones de aprendizaje. Se examinaron los términos proporcionados por los usuarios y se exploraron cuáles corresponden bien a la igualación que han acordado ampliamente significado, qué término tiene significados específicos para grupos pequeños, y qué términos son sinónimos.

- Pennington *et al* [18]. Se analizó y se hicieron explícitas las propiedades del modelo necesarias para que las regularidades sintácticas emerjan en vecores de palabras. El resultado es un nuevo modelo de regresión log-bilineal global que combina las ventajas de los dos modelos principales en la literatura: matriz global factorización y ventana de contexto local métodos. El modelo en cuestión aprovechó eficientemente información estadística mediante la formación sólo en los elementos distintos de cero en una matriz de coocurrencia palabra-palabra, en lugar de en toda la matriz dispersa o en el contexto individual ventanas en un gran corpus. El modelo produce un espacio vectorial con una subestructura significativa, como lo demuestra su desempeño. del 75 % en una tarea reciente de analogía de palabras. Él también supera a los modelos relacionados en tareas de similitud y reconocimiento de entidades nombradas.

2.3. Dataset: Social-EQ

M. B. Cartwright y B. Pardo, en su trabajo titulado "Social-EQ: Crowdsourcing an Equalization Descriptor Map" [3], presentaron un dataset de relevancia para la ecualización automática de audio basándose en descripciones verbales. A continuación, se detallan las características esenciales de este conjunto de datos:

Origen

El dataset fue introducido en la 14ª Conferencia de la Sociedad Internacional para la Recuperación de Información Musical (ISMIR) en 2013.

Contenido

- Formato: .csv
- Número de entradas: 1596
- Detalles por entrada: Descripción de audio (descriptor), idioma, valor de consistencia de las calificaciones y respuestas espectrales coherentes (RSC) en diferentes bandas de frecuencia.

Ejemplo de entradas

- Descriptor: hot, Idioma: English, Consistencia de calificaciones: 8, RSC en 20Hz: 0.797209, ... , RSC en 19682Hz: -1.0730892926286768
- Descriptor: wet, Idioma: English, Consistencia de calificaciones: 8, RSC en 20Hz: 0.481343, ... , RSC en 19682Hz: 1.357464893912528

Calidad del dataset

- Procedencia académica: El dataset proviene de un trabajo de investigación presentado en una conferencia académica reconocida, sugiriendo una recolección y procesamiento de datos riguroso.
- Fiabilidad: La consistencia en las calificaciones indica la confiabilidad de los valores RSC para cada descriptor.
- Aplicabilidad: El dataset es adecuado para sistemas que buscan la ecualización automática de audio basándose en descripciones verbales. Además, su estructuración en diferentes bandas de frecuencia facilita un análisis detallado del impacto de cada descriptor en el espectro audible.

Capítulo 3

Marco Teórico

3.1. Sonido

El sonido es un tipo de energía que se propaga en forma de ondas a través de un medio, como el aire, el agua o los sólidos. Tratándose de ondas, la frecuencia determina el tono y la amplitud de un sonido [14]. Los contextos ambientales pueden influir en la propagación del sonido, con factores como la humedad y la temperatura afectando la velocidad y la absorción del sonido [15].

Aunque un sonido simple, como un tono puro o un sonido de percusión puede ser representado por una sola onda sonora, muchos sonidos complejos están compuestos por múltiples ondas con diferentes frecuencias y amplitudes, las cuales combinan para formar una forma de onda más compleja [16]. Este fenómeno es particularmente evidente en sonidos con múltiples armónicos, como la música y el habla humana, los cuales consisten en una frecuencia fundamental y una serie de armónicos u overtones que contribuyen a la calidad tímbrica o color tonal del sonido [17].

3.2. Ganancia de audio

La ganancia de audio se refiere a la relación entre la amplitud de la señal de entrada y la amplitud de la señal de salida en un sistema de amplificación de audio [18]. Esta relación se mide en decibelios (dB) y se calcula a partir de la diferencia de niveles de presión sonora (SPL) entre la señal de entrada y la señal de salida [19]. Desde un punto de vista económico, la gestión adecuada de la ganancia es crucial en la industria del audio, ya que el sobre-amplificar puede causar daños costosos al equipo y, a la inversa, la falta de ganancia adecuada puede llevar a una experiencia de usuario subóptima [20].

3.3. Filtro de audio

Un filtro de audio es un dispositivo que se utiliza para modificar las características de frecuencia de una señal de sonido, permitiendo ciertas frecuencias para pasar mientras reduce o bloquea otras [21]. Se utilizan comúnmente para eliminar ruido no deseado de una señal, ajustar el balance de frecuencias de un

sonido o para crear efectos especiales [22]. Desde un punto de vista político, la filtración de audio puede ser utilizada en tecnologías para el monitoreo y la censura, ajustando o eliminando ciertos sonidos o palabras [23].

3.3.1. Tipos de filtros

- *Filtro de paso bajo*: Permite el paso de las frecuencias bajas de una señal mientras disminuye las frecuencias más altas [24].
- *Filtro de paso alto*: Permite que los componentes de alta frecuencia de una señal pasen mientras disminuye las frecuencias más bajas [25].
- *Filtro de banda*: Permite seleccionar un rango de frecuencias determinado, llamado banda de paso, para dejarlo pasar mientras disminuye o bloquea las frecuencias que se encuentran fuera de ese rango [26].
- *Filtro de rechazo de banda*: Atenúa una estrecha gama de frecuencias, conocida como la muesca, mientras permite que otras frecuencias pasen a través [27].
- *Filtro de estante*: Permite el aumento o la reducción de las frecuencias por encima o por debajo de una frecuencia de corte determinada [28].

3.4. Ecualización

La ecualización de audio es el proceso de ajustar la respuesta de frecuencia de una señal de audio mediante el aumento o la reducción selectiva de ciertas frecuencias [29]. Esto se puede lograr utilizando filtros de audio, que permiten controlar el nivel de señales de audio en diferentes bandas de frecuencia [30]. Desde un punto de vista económico, la ecualización es esencial en la producción musical y de radio para producir sonidos que sean atractivos para los consumidores y, por ende, rentables [31].

Matemáticamente, la ecualización se puede representar utilizando una función de transferencia, que describe cómo la amplitud de la señal varía en función de la frecuencia [32]. Esta se puede representar como una curva en un gráfico que muestra la amplitud de la señal en decibelios (dB) en el eje vertical y la frecuencia en hertzios (Hz) en el eje horizontal [33]. La forma de la curva refleja cómo se están ajustando las diferentes frecuencias [34].

El objetivo de la ecualización es mejorar la calidad del sonido, corregir problemas de resonancia, eliminar la retroalimentación y mejorar la inteligibilidad del habla en aplicaciones de comunicación, entre otras aplicaciones [35]. La ecualización de audio se puede realizar de forma analógica o digital, y hay una amplia variedad de tipos de ecualización disponibles, incluyendo ecualización paramétrica, gráfica, de bandas, de estantes y de campana [36].

3.5. Ecualización paramétrica

La ecualización paramétrica es una técnica de procesamiento de audio que permite ajustar de manera precisa y flexible la respuesta en frecuencia de una

señal de audio [14]. En lugar de utilizar un filtro con una frecuencia de corte fija, como en el caso de los filtros gráficos, la ecualización paramétrica permite ajustar los parámetros del filtro, como la frecuencia central, el ancho de banda y la ganancia, de forma independiente [15]. Esto permite una mayor precisión y control en la corrección de la respuesta en frecuencia y en la creación de efectos de sonido [16].

Los diagramas de ecualización paramétrica representan de manera gráfica la modificación del espectro de frecuencias de una señal de audio en un sistema de ecualización paramétrica [17]. Generalmente, estos diagramas tienen dos ejes, uno vertical que representa la ganancia en decibelios y otro horizontal que representa la frecuencia en Hertz [18]. En este último eje se pueden ver varias bandas que corresponden a las diferentes frecuencias en las que se realizará la ecualización [19]. En cada banda se pueden ajustar la ganancia, la frecuencia central y el ancho de banda [20].

La ecualización paramétrica en un sistema de audio consiste en modificar la amplitud de las señales en cada banda de frecuencia para realzar o atenuar determinadas características sonoras [21]. Por ejemplo, si se desea realzar los graves de una grabación, se puede utilizar la ecualización paramétrica para aumentar la ganancia en la banda de frecuencias correspondiente a los graves [22]. Este ajuste se realiza a través de un proceso matemático denominado filtrado, en el que cada banda de frecuencia se puede representar como un filtro, y la ganancia, la frecuencia central y el ancho de banda determinan las características de ese filtro [23].

Por tanto, los diagramas de ecualización paramétrica son una herramienta visual que permite ajustar la ecualización de una señal de audio para realzar o atenuar características sonoras específicas [24]. El proceso de ecualización paramétrica se basa en el uso de filtros para modificar la amplitud de las señales en cada banda de frecuencia, y el ajuste de la ganancia, la frecuencia central y el ancho de banda determina las características de cada filtro [25]. La figura 3.1 muestra el gráfico de una ecualización paramétrica [26]. En esta gráfica el eje horizontal representa las frecuencias de la señal de audio. Estas frecuencias se expresan en Hertz (Hz) y abarcan el rango audible del espectro, desde aproximadamente 20 Hz (bajas frecuencias) hasta 20 kHz (altas frecuencias). El eje vertical, por otro lado, representa la ganancia o atenuación aplicada a las frecuencias, generalmente en decibelios (dB). Valores positivos en este eje indican amplificación (aumento de volumen), mientras que valores negativos indican atenuación (reducción de volumen). La curva de ecualización es la línea que muestra cómo se modifica el nivel de diferentes frecuencias, y su forma depende de los ajustes de los filtros aplicados.

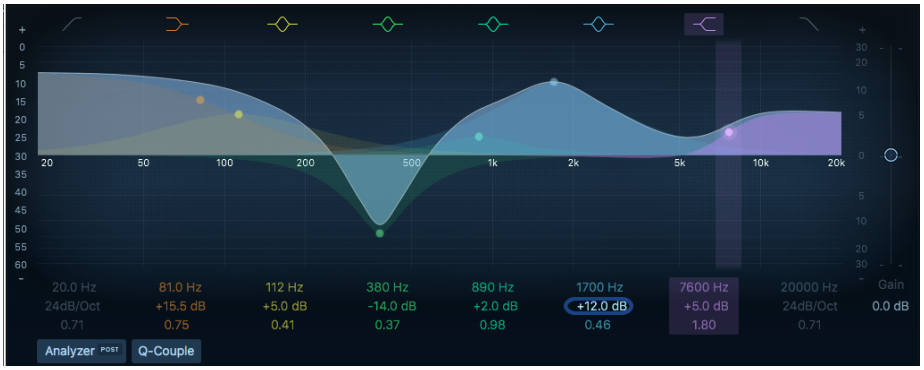

Figura 3.1: Gráfico de una ecualización paramétrica [42].

3.6. Procesamiento del lenguaje natural y modelos utilizados en la ecualización automática

El Procesamiento del Lenguaje Natural (NLP, por sus siglas en inglés) es un campo de estudio que combina la informática, la inteligencia artificial y la lingüística para permitir que las computadoras procesen, entiendan y generen lenguaje humano [20]. Se trata de desarrollar algoritmos y modelos que puedan analizar y derivar significado de grandes cantidades de datos de lenguaje natural, incluyendo texto, habla y gestos. Gracias a los avances en NLP, han surgido aplicaciones que han revolucionado industrias, tales como la búsqueda en la web, análisis de sentimientos en redes sociales y asistentes virtuales como Siri o Alexa.

Dentro del NLP, existen tareas clave que permiten a las máquinas comprender el lenguaje humano, incluyendo la segmentación de texto, el análisis morfológico, el etiquetado de partes de la oración, y el análisis tanto sintáctico como semántico [21]. Estos métodos buscan entender la estructura, el significado y el contexto del lenguaje natural. Además, en el ámbito político, el NLP puede ser una herramienta potente para monitorear opiniones públicas, detectar desinformación y analizar discursos políticos.

3.6.1. Modelos de incrustación de palabras

Los modelos de embedding de palabras son una técnica de procesamiento de lenguaje natural (NLP) que representan las palabras como vectores densos y de baja dimensionalidad en un espacio vectorial continuo [11]. Estos modelos buscan capturar las relaciones semánticas y sintácticas entre las palabras, de modo que las palabras similares sean representadas por vectores similares, mientras que las palabras disimilares sean representadas por vectores disimilares.

Existen varios tipos de modelos de embedding de palabras, incluyendo modelos basados en conteo como Latent Semantic Analysis (LSA) y modelos basados en predicciones como Word2Vec y GloVe [11]. Los modelos basados en conteo utilizan técnicas de factorización matricial para identificar la estructura semántica subyacente de un corpus de texto, mientras que los modelos basados

en predicciones utilizan redes neuronales para predecir el contexto en el que aparece una palabra en una oración.

Una vez que se ha entrenado un modelo de embedding de palabras en un corpus grande de texto, se puede utilizar para una variedad de tareas de NLP, como análisis de sentimiento, traducción de idiomas y clasificación de texto. Al representar las palabras como vectores, estos modelos pueden capturar los matices sutiles del lenguaje y mejorar el rendimiento de los modelos de NLP subsiguientes. Desde un punto de vista ambiental, la construcción y el uso eficiente de estos modelos pueden requerir menos recursos computacionales, lo que resulta en un menor consumo de energía.

Modelo word2vec

Word2vec es un modelo de procesamiento de texto que convierte las palabras en vectores de características. Este modelo funciona a través de una red neuronal de dos capas que toma como entrada un corpus de texto y produce como salida vectores que representan las palabras en ese corpus. Aunque Word2vec no es considerado una red neuronal profunda, su capacidad para transformar el texto en una forma numérica hace posible que otras redes neuronales más complejas puedan comprender el lenguaje natural.

Este modelo utiliza dos estrategias de entrenamiento: la primera es predecir una palabra objetivo utilizando el contexto en el que se encuentra, conocida como continuous bag of words (CBOW), y la segunda es utilizar una palabra para predecir el contexto en el que se encuentra, llamada skip-gram. El modelo de skip-gram produce resultados más precisos en grandes conjuntos de datos, por lo que es la estrategia preferida en este caso. La figura 3.2 muestra la arquitectura de ambas variaciones del modelo word2vec [40].

Formalmente, el modelo CBOW trata de predecir la palabra objetivo w_t usando un contexto de palabras $(w_{t-1}, w_{t-2}, ..., w_{t-k})$. La función objetivo a maximizar se define como:

$$\text{máx} \frac{1}{T} \sum_{t=k+1}^{T-k} \log p(w_t | w_{t-1}, \ldots, w_{t-k})$$

Donde la probabilidad $p(w_t|w_{t-1}, \ldots, w_{t-k})$ se modela usando softmax:

$$p(w_t | w_{t-1}, \ldots, w_{t-k}) = \frac{\exp(v_{w_t}^T \cdot h)}{\sum_{w=1}^{W} \exp(v_w^T \cdot h)}$$

Aquí, v_w es el vector de palabras y h es la media de los vectores de contexto. Por otro lado, el modelo skip-gram trata de predecir el contexto $(w_{t-1}, w_{t-2}, ..., w_{t-k})$ usando la palabra objetivo w_t. La función objetivo a maximizar en este caso es:

$$\text{máx} \frac{1}{T} \sum_{t=1}^{T} \sum_{-k \leq j \leq k, j \neq 0} \log p(w_{t+j} | w_t)$$

Donde la probabilidad $p(w_{t+j}|w_t)$ se modela usando softmax como en CBOW.

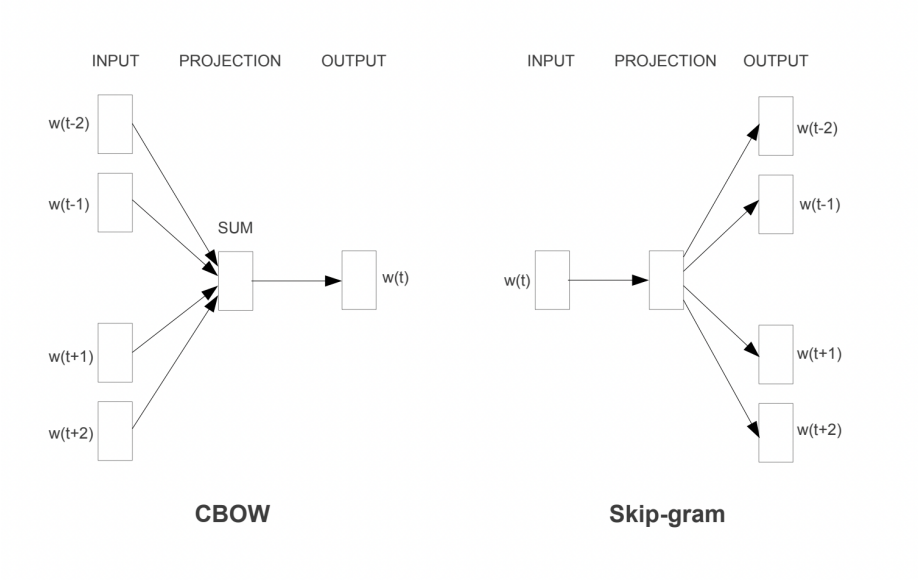

Figura 3.2: Arquitecturas CBOW y Skip-gram [42].

Modelo tok2vec

El modelo tok2vec de la librería spaCy de Python es una poderosa herramienta para la generación de representaciones vectoriales densas de tokens. Este modelo utiliza varias capas de transformaciones lineales y funciones de activación no lineales para procesar una secuencia de tokens de entrada. Los tokens se convierten en vectores one-hot y luego se multiplican con una matriz de peso [41].

Formalmente, considere un token t_i que se convierte en una representación one-hot x_i. La primera operación es una transformación lineal multiplicando x_i por una matriz de peso W:

$$h_i^{(1)} = W \cdot x_i$$

Esta transformación lineal es seguida por una función de activación no lineal, como ReLU:

$$a_i^{(1)} = \text{ReLU}(h_i^{(1)})$$

Esta operación se puede repetir para varias capas:

$$h_i^{(l+1)} = W^{(l)} \cdot a_i^{(l)}$$

$$a_i^{(l+1)} = \text{ReLU}(h_i^{(l+1)})$$

Finalmente, cada token está representado por un vector de longitud fija, que podría pasar por otras capas adicionales o usarse directamente como entrada para tareas de procesamiento de lenguaje natural, como análisis de sentimientos, reconocimiento de entidades nombradas, o traducción automática. Este vector

captura las complejas relaciones entre los tokens y su contexto, produciendo una representación rica y densa del significado de cada token.

Modelo GloVe

El modelo GloVe es un algoritmo de aprendizaje no supervisado que se utiliza para generar representaciones vectoriales de palabras a partir de grandes cantidades de datos de texto. En lugar de simplemente contar la frecuencia de co-ocurrencia de las palabras, el modelo utiliza una matriz de co-ocurrencia de palabras para calcular la probabilidad de que dos palabras aparezcan juntas en un contexto de ventana fija. Luego, se utilizan técnicas de álgebra lineal para ajustar las representaciones vectoriales de las palabras para reflejar su relación semántica y sintáctica[39].

Formalmente, sea X la matriz de coocurrencia entre palabras, donde X_{ij} representa la cantidad de veces que la palabra j ocurre en el contexto de la palabra i. Sea $X_i = \sum_k X_{ij}$ el número de veces que cualquier palabra aparece en el contexto de la palabra i. Finalmente, sea $P_{ij} = P(j|i) = X_{ij}/X_i$ la probabilidad de que la palabra j aparezca en el contexto de la palabra i. Ahora bien, considerando las palabras i y j, se puede estudiar la relación entre estas estudiando el ratio de sus probabilidades de co-ocurrencia con las palabras k. De esta forma, para palabras k relacionadas a j pero no a i, se espera que el ratio P_{ik}/P_{jk} sea grande. Similarmente, para palabras relacionadas a i pero no a j, se esperan valores pequeños y para palabras relacionadas con ambas o con ninguna, se esperan valores cercanos a 1. De esta forma, resulta sencillo discriminar palabras poco relevantes para el contexto, lo que sugiere que estas relaciones de probabilidad pueden servir de mejor punto de inicio para el aprendizaje de vectores de palabras que las mismas probabilidades de co-ocurrencia.

Tendiendo en cuenta que la relación P_{ik}/P_{jk} depende de las palabras i, j y k, el modelo se puede expresar de la forma,

$$F(w_i, w_j, w_k) = \frac{P_{ik}}{P_{kj}}.$$

Donde sea $d \in \mathbb{N}$, $w \in \mathbb{R}^d$ y F una función que codifique P_{ik}/P_{jk}. Note que como los espacios vectoriales son inherentemente estructuras lineales, se puede codificar la presencia de w_i y w_j como $w_i - w_j$.

$$F(w_i - w_j, w_k) = \frac{P_{ik}}{P_{kj}}$$

Note que mientras los argumentos de F son vectores, la parte derecha de la ecuación 3.2 es un valor escalar. Por lo tanto, para simplificar la parametrización de F, se procede aplicando el producto punto entre los vectores de la siguiente manera,

$$F((w_i - w_j)^T w_k) = \frac{P_{ik}}{P_{kj}}$$

Note que F se puede expresar como un homomorfismo entre los grupos $(\mathbb{R}, +)$ y $(\mathbb{X}, \times)$. Por lo que,

$$F((w_i - w_j)^T w_k) = \frac{F(w_i^T w_k)}{F(w_j^T w_k)}$$

Por (3.2) se obtiene

$$F(w_i^T w_k) = P_{ik} = \frac{X_{ik}}{X_i}$$

La solución a (3.4) es $F = \exp$, por lo tanto,

$$w_i^T w_k = \log(P_{ik}) = \log(X_{ik}) - \log(X_i)$$

Note que para evitar el problema de divergencia con el logaritmo, es necesario cambiar la expresión $\log(X_{ik})$ por $\log(1 + X_{ik})$. Adicionalmente, como $\log(X_i)$ no depende de k, se puede abstraer su valor por un bias b_i. A su vez, añadiendo un bias adicional para w_k, se logra expresar el modelo como:

$$w_i^T w_k + b_i + b_k = \log(X_{ik})$$

3.6.2. Aprendizaje de máquina y modelos utilizados en la ecualización automática de audio

Redes neuronales profundas (DNNs)

Las DNNs son un tipo de redes neuronales artificiales que se inspiran en la estructura y el funcionamiento de las redes neuronales biológicas [5]. Estas redes están compuestas por múltiples capas de nodos interconectados, donde cada capa procesa y transmite información a la siguiente. El elemento básico de las DNNs es la neurona artificial. Las capas se organizan en una estructura de entrada-salida, y las capas intermedias se denominan capas ocultas. Cuanto más profunda sea la red, más complejas pueden ser las relaciones que se pueden modelar en los datos.

El entrenamiento de las DNNs se realiza a través de un proceso llamado retropropagación, que implica ajustar los pesos de las conexiones entre las neuronas para minimizar el error entre las predicciones de la red y los valores reales deseados [5].

Algoritmo del descenso de gradiente

El algoritmo de descenso de gradiente es un método de optimización que busca minimizar una función de manera iterativa [7]. Es ampliamente utilizado en aprendizaje automático y aprendizaje profundo para entrenar modelos al disminuir la función de error (o pérdida).

La regla de actualización general para el descenso de gradiente es la siguiente:

$$\theta = \theta - \alpha * \delta j(\theta) \tag{3.1}$$

Esta ecuación describe cómo se actualizan los parámetros del modelo en cada iteración del algoritmo. Los componentes de la ecuación son:

- θ: Representa los parámetros del modelo (como los pesos y sesgos) que se están ajustando para minimizar la función de pérdida.

- α: Es la tasa de aprendizaje, un hiperparámetro que controla la velocidad a la que el algoritmo aprende. Un valor de más pequeño implica que el algoritmo aprende más lentamente, mientras que un valor más grande puede acelerar el aprendizaje pero también puede provocar oscilaciones o divergencia.
- $\delta j(\theta)$: Es el gradiente de la función de pérdida $J(\theta)$.

Regresion logística

La regresión logística es un método estadístico utilizado para analizar y predecir resultados binarios en un conjunto de datos. Es un tipo de modelo lineal generalizado (GLM) que utiliza la función logística para modelar la probabilidad de una variable de respuesta binaria [7].

La regresión logística se utiliza comúnmente en el aprendizaje automático y el análisis de datos para tareas de clasificación, como detección de spam, diagnóstico médico y predicción de abandono de clientes [2].

La función logística, también conocida como función sigmoide, se describe como:

$$P(Y = 1|X) = \frac{1}{1 + e^{-z}} \tag{3.2}$$

donde $z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_n X_n$, X representa las características de entrada y $\beta_0, \beta_1, \beta_n$ son parámetros del modelo [7].

Capítulo 4

Implementación

4.1. Dataset: Social EQ

El conjunto de datos SocialEQ, fundamental para este estudio, agrupa descriptores semánticos asociados a configuraciones de ecualización en el ámbito del audio. Cada registro en este conjunto abarca un descriptor semántico, el idioma del mismo, identificador de audio, una puntuación de coherencia, y valores correspondientes a 40 parámetros de ecualización.

Los participantes aportaban términos descriptivos en su idioma preferido y seleccionaban un archivo de audio a ser modificado mediante un plugin de ecualización. El proceso incluía la presentación de 40 variaciones del archivo de audio, cada una con una configuración de ecualización distinta. Se pedía a los participantes calificar la adecuación del término seleccionado con el sonido modificado. Este método incluyó repeticiones de ciertas variaciones para comprobar la coherencia en las respuestas, evaluada a través de la correlación de Pearson entre calificaciones de pruebas y ejemplos repetidos.

Los resultados contribuyeron al desarrollo de ajustes relativos de ecualización para 40 bandas de frecuencia distintas, basados en la percepción y evaluación de los usuarios. Este dataset contiene 1595 muestras, pero se restringió la investigación a descriptores en inglés, lo cual redujo el número a 918 ejemplos, representando 388 descriptores únicos.

4.2. División de Datos para Entrenamiento y Pruebas: Estrategias y Aplicaciones

En este estudio, se planteó una hipótesis clave: que una capa de incrustación de palabras mejoraría la capacidad del modelo para predecir configuraciones de parámetros de ecualización (EQ) a partir de descriptores semánticos previamente no vistos. Para validar esta hipótesis, se diseñó un esquema de validación cruzada de cuatro pliegues, con el objetivo de garantizar que las palabras en el conjunto de prueba fueran completamente desconocidas para el modelo durante su fase de entrenamiento.

4.2.1. Selección de Palabras para la Validación Cruzada

Se recopiló un conjunto de descriptores semánticos frecuentemente utilizados en la literatura de mezcla de audio, a los que se denominó palabras de Alta Calidad (HQ). Estas palabras fueron seleccionadas objetivamente de fuentes literarias y se complementaron con descriptores pertenecientes a una ontología jerárquica. Además, se identificaron palabras Altamente Calificadas (HR), las cuales, si bien no necesariamente tienen un significado semántico intrínseco, poseen una alta coherencia en el conjunto de datos SocialEQ, con una puntuación de coherencia superior a 0.7. Esto indica una fuerte asociación del usuario con una configuración específica de EQ.

4.2.2. Configuración de la Validación Cruzada

Cada uno de los cuatro pliegues de validación contenía una mezcla de palabras HQ y HR, distribuidas de tal manera que cada palabra fuera evaluada al menos una vez. Se aseguró que no hubiera superposición entre los conjuntos de entrenamiento y prueba, es decir, las palabras en el conjunto de prueba no aparecieron en ningún momento en el conjunto de entrenamiento correspondiente. Esto permitió que el modelo fuera evaluado exclusivamente en su capacidad de generalizar a partir de descriptores semánticos desconocidos.

El modelo se entrenó y evaluó cuatro veces, una por cada pliegue, y se reportó el rendimiento promedio de estas ejecuciones. Este enfoque metodológico permitió una evaluación objetiva de la hipótesis planteada, verificando la efectividad de la capa de incrustación de palabras en la predicción de configuraciones de EQ para nuevos descriptores semánticos.

Fold 1	Fold 2	Fold 3	Fold 4
smooth, muffled, crisp, punch, clean, brittle, muddy, soothing, clear, brassy, caring, mellow, throbbing, cooing, fluffy, good, excited, squeaking, punchy, funky, whispered, disgusting, beautiful, reserved, serene, thumpy, pleasurable, whispering, gentle, energetic, peace	**crunchy, woody, flat, metallic, dull, tinny, cold, booming, deep,** energizing, heart-warming, edgy, heavy, edge, strong, enchanting, cheerful, plodding, quiet, radiant, biting, brass, pleasing, light, taco, gruff, exciting, love, heat, techno, solemn	**sweet, warm, airy, full, boxy, bright, boom, fat, shrill,** calm, velvety, hard, rich, noisy, down, rumble, sloppy, relaxing, peaceful, romantic, low, hot, thunderous, frigid, happy, poor, cool, tense, jagged, forceful, aggressive	**sharp, big, dark, hollow, harsh, smooth, muffled, crisp, punch,** mournful, clarity, genius, bold, twangy, soft, splash, slow, wistful, brash, fancy, cute, rousing, loud, breezy, large, passionate, baseball, huge, icy, brassy, caring

Figura 4.1: Cuatro pliegues de validación cruzada del conjunto de datos. Las palabras de prueba de cada pliegue se presentan en la tabla. Para cada pliegue, el conjunto de entrenamiento consiste en palabras que no están en el conjunto de prueba.

4.2.3. Implementación Adicional para el Producto Final

Además de la configuración de validación cruzada, se implementó una versión adicional del modelo utilizando la función `train_test_split` de `scikit-learn`. Este enfoque permitió maximizar el uso de los datos disponibles, creando un conjunto de entrenamiento más grande y permitiendo que el modelo final estuviera

mejor ajustado y optimizado para su uso con usuarios finales en un entorno de producción.

El `train_test_split` se utilizó para dividir el conjunto de datos en un 80 % para entrenamiento y un 20 % para pruebas, asegurando así que el modelo fuera expuesto a la mayor cantidad de datos posible durante el entrenamiento, mejorando su capacidad de generalización en el producto final. Esta estrategia fue especialmente útil para mejorar la robustez y precisión del modelo en un escenario real, donde la diversidad y cantidad de datos disponibles pueden ser limitadas.

La implementación de esta versión final del modelo con `train_test_split` fue crucial para garantizar que el producto entregado a los usuarios finales tuviera un rendimiento superior y estuviera preparado para manejar una variedad de entradas semánticas en condiciones del mundo real.

4.3. Modelos de Incrustación de Palabras

En la implementación del proyecto se analizó el impacto de diversas técnicas de incrustación de palabras en la capacidad predictiva del modelo en relación a los ajustes de parámetros de ecualización a partir de descriptores semánticos. Se partió de la base de que una representación vectorial adecuada de las palabras es esencial para el procesamiento por parte de la red neuronal, especialmente en términos que no formaron parte del conjunto de entrenamiento.

El vocabulario del modelo incluye el total de palabras que la red es capaz de interpretar, las cuales se traducen en vectores mediante técnicas de incrustación. Tradicionalmente, una palabra se convierte en un vector de representación unitaria o *one-hot encoding*, donde el tamaño del vector corresponde al número de palabras únicas en el vocabulario, en este caso, 388 términos del conjunto de datos SocialEQ.

Para superar las limitaciones que presenta el *one-hot encoding*, donde cada par de palabras se encuentra a la misma distancia euclidiana, se implementaron modelos de incrustación más avanzados, que permiten representar las palabras en un espacio vectorial de menor dimensionalidad, donde la red neuronal puede diferenciar entre palabras con similitudes o diferencias semánticas.

Se integraron cuatro modelos de incrustación de palabras:

- **GloVe (Global Vectors for Word Representation)**: Es un algoritmo de aprendizaje no supervisado para obtener representaciones vectoriales de palabras, que destaca por capturar las relaciones estadísticas de coocurrencia en grandes corpus de texto. Se utilizó la variante de GloVe entrenada con 840 mil millones de tokens y un vocabulario de 2.2 millones de términos.

- **Tok2Vec**: Proporcionado por spaCy, este modelo se especializa en la generación de representaciones vectoriales densas de tokens, diseñado para captar las sutilezas contextuales y semánticas del lenguaje.

- **GPT (Generative Pretrained Transformer)**: Un modelo de lenguaje preentrenado que sobresale por su habilidad para capturar y generar texto basándose en contextos amplios, ofreciendo una comprensión profunda del lenguaje y las relaciones semánticas complejas.

- **BERT (Bidirectional Encoder Representations from Transformers)**: Similar a GPT en su naturaleza preentrenada y capacidad contextual, BERT utiliza un enfoque bidireccional para comprender el contexto, capturando la información de las palabras basándose en las que se encuentran antes y después en la oración.

Estos modelos no solo representan un avance sobre los métodos tradicionales de incrustación de palabras sino que también aportan una nueva perspectiva en el tratamiento de descriptores semánticos, especialmente aquellos que no son comunes en la literatura de mezcla de audio. La implementación de GPT y BERT, que no fueron utilizados en el trabajo de referencia, ofreció una oportunidad única para explorar la eficacia de estos modelos en la comprensión contextual del lenguaje y su influencia en la predicción de configuraciones de EQ.

4.4. Arquitectura de Aprendizaje Automático

4.4.1. Capa de Incrustación de Palabras

Se evaluaron cuatro modelos preentrenados de incrustaciones de palabras: GloVe, Tok2Vec, GPT y BERT. Estos modelos representan palabras en vectores semánticos de 300 dimensiones, lo cual estandariza la entrada para la red neuronal. La conversión inicial de palabras a una representación codificada en caliente (*one-hot encoded*) es seguida por su transformación en vectores semánticos mediante una matriz de incrustación. Dicha matriz, cuyos pesos están congelados para evitar el entrenamiento debido al tamaño limitado del conjunto de datos, actúa como una capa no entrenable en la red.

4.4.2. Arquitectura de la Red Neuronal

La red neuronal (Figura 4.2) está diseñada para mapear las incrustaciones de palabras a predicciones de parámetros de ecualización. Esta tarea requiere una red de cierta profundidad para aprender la traducción entre representaciones lingüísticas y parámetros técnicos. La red se compone de varias capas completamente conectadas, detalladas como sigue:

- **Capa de Entrada**: La entrada a la red es un tensor de tipo entero (*int64*) de forma (1,), representando secuencias de índices de palabras.
- **Capa de Incrustación**: Se transforman las secuencias de índices en vectores semánticos usando la capa de incrustación previamente definida.
- **Procesamiento de la Secuencia**: La dimensión adicional se elimina usando una capa Lambda que aplica la función *squeeze* de TensorFlow.
- **Capas Densas**: La red incluye sucesivas capas densas con 300, 200, 100, 80 y 60 unidades, respectivamente. Cada una de estas capas aplica una activación ReLU y un *dropout* de 0.05 para regularización.
- **Capa de Salida**: Finalmente, se conecta a una capa de salida con 40 unidades, utilizando una activación *sigmoid* para predecir los parámetros de ecualización.

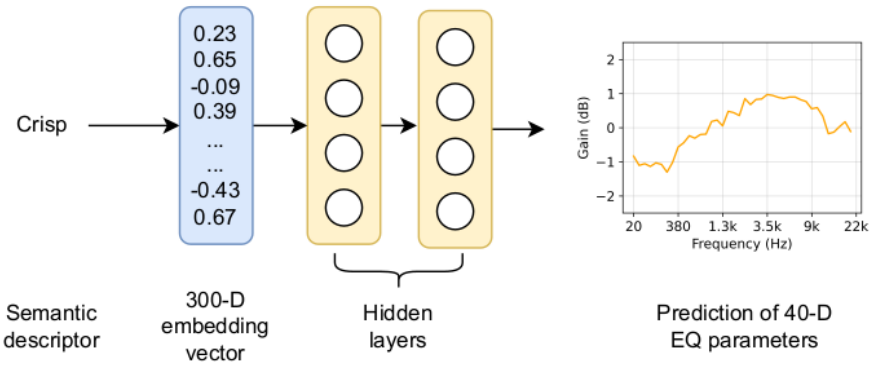

Figura 4.2: Un diagrama esquemático de cómo la red aprende una traducción de descriptores semánticos a parámetros de EQ. Fuente [11]

El modelo se compila con una función de pérdida de error absoluto medio y un optimizador SGD con una tasa de aprendizaje. Las métricas incluyen el error porcentual medio absoluto para evaluar el rendimiento. Este enfoque de modelado busca optimizar la precisión en la predicción de parámetros de ecualización a partir de las incrustaciones de palabras, adaptándose a las características específicas del conjunto de datos y la tarea en cuestión.

4.4.3. Normalización de los Parámetros de Ecualización

En el tratamiento de los datos para la red neuronal, se descartó la normalización mínima-máxima tradicional que calcula los valores máximos y mínimos en el conjunto de entrenamiento. Esto se debe a que, si existen valores atípicos entre los datos del conjunto de prueba, ciertas características podrían verse amplificadas o disminuidas indebidamente. Además, dado que se predicen valores para 40 bandas de ecualización, este problema se vuelve más crítico. Por tanto, se fijaron valores mínimos y máximos para cada parámetro de EQ en -4 dB y +4 dB, respectivamente. En otras palabras, el corte/impulso máximo dentro de cada banda de EQ fue de 4 dB. Los valores se normalizaron linealmente al rango de 0 a 1, de modo que -4 dB correspondería a 0 y +4 dB a 1 en la capa de salida.

4.4.4. Capa de Salida y Función de Pérdida

La capa de salida de la red contiene 40 neuronas, cada una prediciendo un valor para una banda de EQ. Como los datos se normalizaron en el rango de 0 a 1, se utilizaron funciones de activación sigmoideas para las neuronas de salida. Dado que se aborda un problema de regresión, se adoptó la función de pérdida de error absoluto medio, la cual es comúnmente utilizada en muchos estudios. Todas las bandas de EQ recibieron la misma importancia al promediar el error para la función de pérdida. La red se entrenó utilizando el descenso de gradiente estocástico con una tasa de aprendizaje inicial de 0.1. La tasa de aprendizaje se ajustó en un factor de 0.96 tras cada 10,000 actualizaciones de pesos.

4.4.5. Ensamble de Modelos de Incrustación de Palabras

En la implementación teórica del proyecto, se construyó un ensamble de modelos de incrustación de palabras para enriquecer la representación semántica de los términos y mejorar la predicción de parámetros de ecualización en bandas de frecuencia. Este ensamble combinó las fortalezas individuales de GloVe, Tok2Vec, GPT y BERT. GloVe aportó un conocimiento basado en la co-ocurrencia estadística, mientras que Tok2Vec agregó una comprensión contextual ajustada por el uso específico de palabras. BERT, por su parte, contribuyó con su comprensión bidireccional de las palabras en contexto, y GPT ofreció una capacidad generativa basada en predicciones de secuencias largas.

Cada vector de incrustación de palabras de los modelos individuales se concatenó para formar un único vector de alta dimensionalidad. Esta representación combinada captura una imagen más rica y matizada del significado de las palabras, lo que se espera que resulte en una predicción más precisa de las configuraciones de EQ.

La concatenación de incrustaciones de diferentes modelos aprovecha la información semántica única que cada uno ofrece. Por ejemplo, GloVe proporciona representaciones basadas en estadísticas de co-ocurrencia de palabras, mientras que BERT y GPT aportan un contexto más amplio y comprensión del lenguaje a nivel de secuencia. Concatenar estos vectores permite que la red neuronal capte una gama más amplia de relaciones semánticas y sutilezas lingüísticas que no se capturarían usando un solo modelo de incrustación. Esta rica representación vectorial integrada posibilita que la red aprenda y haga predicciones más precisas y matizadas, fundamentales para tareas como la predicción de parámetros de EQ en procesamiento de señales de audio.

4.4.6. Capa de Atención

En la arquitectura del proyecto se incluyó una capa de atención para cada tipo de incrustación de palabras. La capa de atención mejora el modelo al permitirle centrarse en partes más relevantes de las incrustaciones de palabras al realizar la predicción de parámetros. Esto es crucial para captar las sutilezas y la importancia relativa de diferentes aspectos dentro de las incrustaciones de alta dimensionalidad. Por ejemplo, la atención puede permitir que el modelo distinga entre las connotaciones sutiles de palabras con múltiples significados o la relevancia de ciertos aspectos del contexto en las incrustaciones de BERT o GPT.

Concatenar los resultados de las capas de atención permite al modelo integrar y aprovechar eficazmente la información enfocada de todas las representaciones de incrustación, lo que puede ser particularmente valioso en tareas complejas como la ecualización de audio, donde diferentes aspectos de las palabras pueden tener diferentes grados de influencia en los parámetros de salida.

4.5. Herramienta de Ecualización Automática

Se implementó un sistema escalable y robusto para la ecualización automática de audio que se ejecuta desde la consola. Este sistema, diseñado para optimizar las tareas de procesamiento de audio mediante el uso de descritores

semánticos, acepta diversos parámetros que facilitan su ejecución y personalización según las necesidades del usuario. Los parámetros incluyen el archivo de audio de entrada, el archivo de salida, el modelo de embedding a utilizar y la palabra descriptiva que guiará la ecualización.

4.5.1. Descripción del Sistema

El sistema desarrollado recibe los siguientes parámetros:

- **Path al audio de entrada (.wav)**: La ruta del archivo de audio que será sometido al proceso de ecualización.
- **Path de salida**: La ruta donde se guardará el archivo de audio ecualizado.
- **Modelo de embedding a utilizar**: La elección del modelo de embedding (*"gpt", "bert", "tok2vec", "word2vec", .*^{en}*semble"*) que proporcionará las representaciones vectoriales del descriptor semántico.
- **Palabra descriptiva**: Un término que describe la configuración de ecualización deseada.

A continuación se detalla el funcionamiento y las características técnicas del sistema.

4.5.2. Flujo de Ejecución del Sistema

El flujo de trabajo del sistema consta de varias etapas:

1. **Carga y Validación de Parámetros**: El sistema inicia leyendo y validando los parámetros de entrada. Se asegura de que los paths de los archivos sean accesibles y que el modelo de embedding solicitado sea uno de los disponibles. Si algún parámetro es inválido, el sistema genera mensajes de error específicos y termina su ejecución.
2. **Carga del Audio de Entrada**: Una vez validados los parámetros, el sistema lee el archivo de audio de entrada utilizando librerías especializadas que garantizan la correcta lectura y manejo del archivo.
3. **Preprocesamiento del Audio**: En esta etapa, se realiza un preprocesamiento del audio para asegurar que esté en el formato adecuado para el análisis, incluyendo la normalización de la señal y la eliminación de ruidos no deseados.
4. **Vectorización del Descriptor Semántico**: El sistema convierte el descriptor semántico en un vector utilizando el modelo de embedding seleccionado. Estos modelos, como GloVe, Tok2Vec, GPT y BERT, permiten obtener representaciones vectoriales contextuales que capturan características semánticas detalladas.
5. **Predicción de Parámetros de EQ**: Utilizando el vector generado, la red neuronal predice los ajustes de EQ correspondientes. La arquitectura de la red incluye capas completamente conectadas que traducen estas incrustaciones en configuraciones específicas de EQ.

6. **Aplicación de EQ al Audio**: Los parámetros de ecualización calculados se aplican al audio mediante algoritmos de procesamiento digital de señales (DSP). Este proceso ajusta las amplitudes de distintas bandas de frecuencia según los valores predichos, generando así una salida de audio ecualizada que refleja fielmente el descriptor semántico proporcionado.

7. **Exportación del Audio Modificado**: Finalmente, el audio modificado se guarda en la ruta de salida especificada. Se asegura que el archivo resultante mantenga alta calidad y su formato sea compatible con aplicaciones estándar de reproducción de audio.

4.5.3. Características Avanzadas del Sistema

Para mejorar la usabilidad y la robustez del sistema, se incluyeron características avanzadas:

- **Generación de Logs**: El sistema genera logs detallados de cada paso del proceso, incluyendo la carga de archivos, validación de parámetros, resultados de predicción y errores encontrados. Estos logs son esenciales para la depuración y el mantenimiento del sistema.

- **Manejo de Errores**: Se implementaron mecanismos de manejo de errores que permiten identificar y resolver problemas rápidamente. Cuando ocurre un error, el sistema proporciona un mensaje claro y específico al usuario.

- **Reversión de Operaciones**: En caso de que ocurra un error durante la ejecución, el sistema puede revertir cualquier cambio realizado hasta ese punto, asegurando que no haya modificaciones parciales en los archivos de salida.

- **Optimización de Recursos**: El sistema está diseñado para ser eficiente en términos de uso de memoria y tiempo de procesamiento. Se utilizaron técnicas como la carga diferida y la liberación de memoria después de cada etapa de procesamiento para asegurar un rendimiento óptimo.

- **Escalabilidad y Flexibilidad**: La arquitectura del sistema permite la inclusión de nuevos modelos de embedding y algoritmos de DSP sin necesidad de realizar cambios significativos en el código existente. Esta flexibilidad facilita futuras mejoras y permite que el sistema se adapte a nuevas tecnologías y métodos.

4.5.4. Ejemplo de Uso

A continuación se muestra un ejemplo de cómo ejecutar el sistema:

```
python auto_eq_script.py --input_path="ruta/al/audio_de_entrada.wav" \
                         --output_path="ruta/al/audio_de_salida.wav" \
                         --model="bert" \
                         --descriptor="brillante"
```

Este comando aplicará una configuración de ecualización al archivo especificado que refleje el descriptor semántico "brillante", utilizando el modelo de embedding BERT para generar las representaciones vectoriales.

Capítulo 5

Resultados y Evaluación

5.1. Introducción

En este capítulo, presentamos los resultados obtenidos de la evaluación de varios modelos en la tarea de ecualización automática de audio utilizando diferentes embeddings preentrenados. El objetivo es analizar el rendimiento de estos modelos, comparar sus errores de entrenamiento y validación, y extraer conclusiones sobre su efectividad en la obtención de la ecualización deseada basada en descripciones semánticas. Las métricas clave evaluadas incluyen el error de entrenamiento, el error de validación y su progresión a lo largo de múltiples épocas [11]. Además, se incluye feedback del uso del sistema de ecualización automático.

5.2. Resultados por Modelo

5.2.1. Modelo Tok2Vec

El modelo Tok2Vec implementa una red neuronal que utiliza embeddings tokenizados específicos para el análisis de texto. La Figura 5.1 muestra el error en el entrenamiento y la validación del modelo Tok2Vec a lo largo de las épocas [3].

Análisis del Modelo

La gráfica de pérdida sugiere que el error de entrenamiento disminuye significativamente al principio, indicando una fase de aprendizaje eficiente. Sin embargo, el error de validación comienza a estabilizarse y luego aumenta ligeramente después de cierto punto, lo que podría indicar que el modelo está experimentando sobreajuste después de aprender bien los patrones iniciales de los datos de entrenamiento [7].

Representaciones de Embeddings

El modelo Tok2Vec utiliza embeddings tokenizados directamente, los cuales están diseñados para capturar representaciones significativas de los tokens en relación con el corpus de datos de entrenamiento. A diferencia de GloVe, BERT

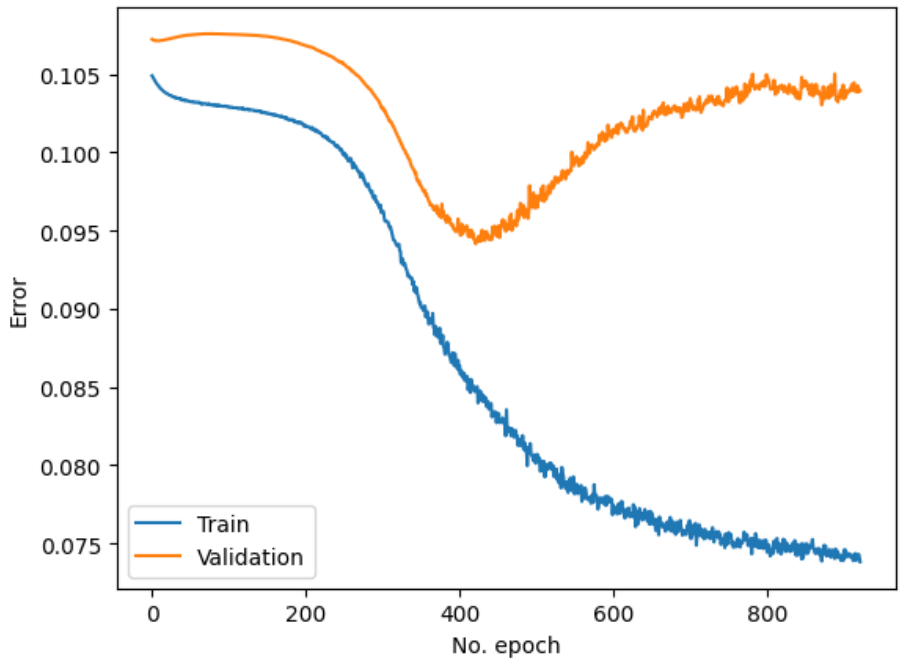

Figura 5.1: Error de Entrenamiento y Validación para el Modelo Tok2Vec durante 900 épocas.

o GPT, que proporcionan embeddings preentrenados ricos en contexto, Tok2Vec se basa en una tokenización específica seguida de capas densas para aprender las representaciones:

- **Tok2Vec:** Utiliza una tokenización directa, lo que permite captar representaciones específicas del conjunto de datos y tarea. Esta tokenización ayuda a aprender relaciones específicas entre los tokens que pueden ser cruciales para la tarea de predicción [2].

Métricas de Evaluación

El rendimiento del modelo Tok2Vec se evalúa usando el error absoluto medio (MAE) y el error porcentual absoluto medio (MAPE). Estas métricas proporcionan una indicación clara de la precisión del modelo al predecir las etiquetas esperadas:

- **Error Absoluto Medio (MAE):** Mide el promedio de los errores absolutos entre las predicciones del modelo y las verdaderas etiquetas [4].

- **Error Porcentual Absoluto Medio (MAPE):** Mide el promedio del error absoluto en términos porcentuales, proporcionando una perspectiva relativa del error con respecto a las verdaderas etiquetas [7].

5.2.2. Modelo GloVe

En este experimento, utilizamos únicamente los embeddings de GloVe para entrenar el modelo. La Figura 5.2 muestra el error en el entrenamiento y la validación del modelo GloVe a lo largo de las épocas [38].

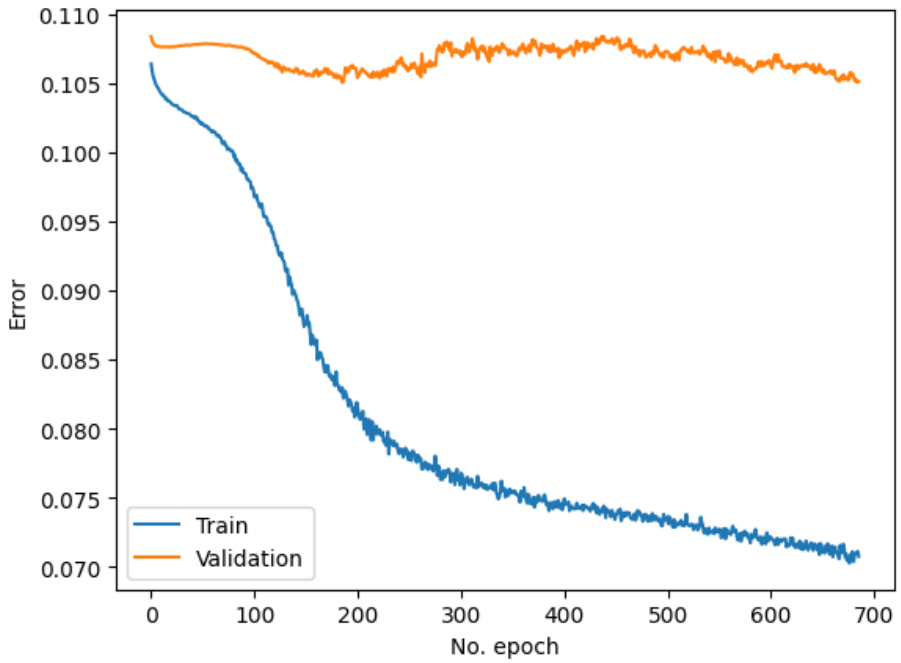

Figura 5.2: Error de Entrenamiento y Validación para el Modelo GloVe durante 700 épocas.

Análisis del Modelo

La gráfica de pérdida ilustra que el error de entrenamiento disminuye significativamente al inicio, lo que indica que el modelo está aprendiendo efectivamente. Sin embargo, el error de validación se estabiliza y empieza a exhibir fluctuaciones, lo cual puede indicar la presencia de sobreajuste. Este comportamiento sugiere que el modelo podría estar ajustándose demasiado a los datos de entrenamiento, afectando su capacidad para generalizar sobre los datos de validación [8].

Representaciones de Embeddings de GloVe

Los embeddings de GloVe son vectores preentrenados que capturan el contexto global de las palabras mediante la estadística de co-ocurrencias. A diferencia de otros métodos, GloVe proporciona representaciones semánticas sólidas que son especialmente buenas para capturar relaciones entre palabras a gran escala [38].

- **GloVe:** Los embeddings de GloVe se construyen aprovechando la matriz de co-ocurrencia de palabras en un corpus extenso. De esta forma, cada palabra se representa como un vector en un espacio de alta dimensionalidad, donde las distancias reflejan similitudes semánticas [38].

Métricas de Evaluación

Evaluamos el rendimiento del modelo GloVe utilizando el error absoluto medio (MAE) y el error porcentual absoluto medio (MAPE), proporcionando una visión clara de la precisión del modelo en la predicción de etiquetas esperadas:

- **Error Absoluto Medio (MAE):** Permite medir la magnitud promedio de los errores entre las predicciones del modelo y los valores reales [4].
- **Error Porcentual Absoluto Medio (MAPE):** Proporciona una perspectiva relativa del error, expresado como un porcentaje, en relación con las verdaderas etiquetas [7].

5.2.3. Modelo BERT

En este experimento, utilizamos los embeddings proporcionados por BERT para entrenar el modelo. La Figura 5.3 muestra el error en el entrenamiento y la validación del modelo BERT a lo largo de las épocas [40].

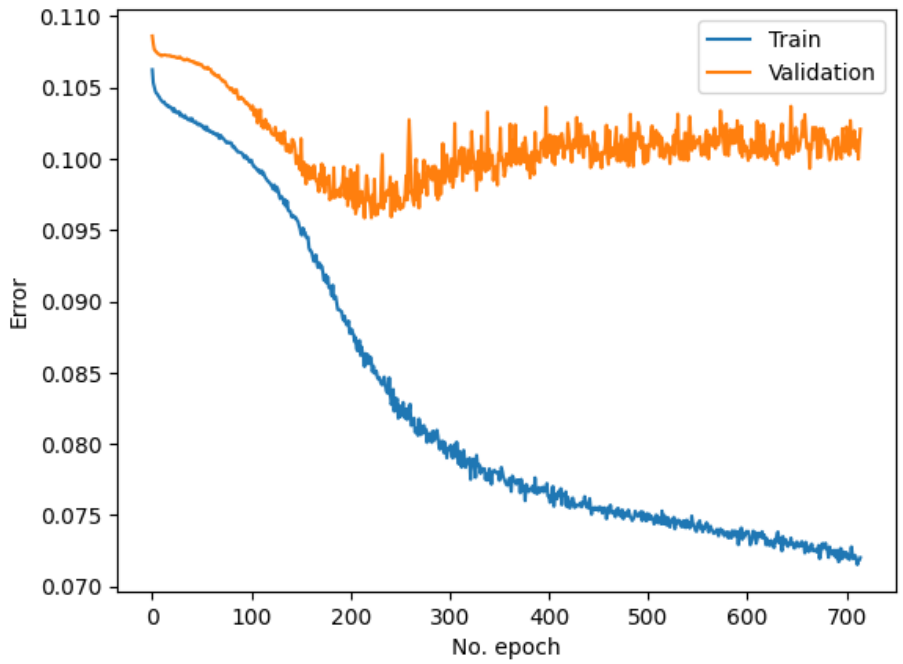

Figura 5.3: Error de Entrenamiento y Validación para el Modelo BERT durante 700 épocas.

Análisis del Modelo

La gráfica de pérdida indica que el error de entrenamiento disminuye considerablemente al inicio, lo que sugiere una etapa de aprendizaje efectiva. Sin embargo, el error de validación se estabiliza y presenta fluctuaciones, lo que puede indicar sobreajuste. Esta tendencia sugiere que el modelo podría beneficiarse de técnicas de regularización adicionales para mejorar su capacidad de generalización [11].

Representaciones de Embeddings BERT

Los embeddings de BERT son vectores contextuales preentrenados que capturan el significado de las palabras en función de su contexto bidireccional. Esto permite que BERT genere representaciones ricas y detalladas de las palabras en diversas oraciones [40].

- **BERT:** Utiliza una arquitectura de transformador para producir embeddings que consideran tanto el contexto a la izquierda como a la derecha de una palabra. Esto resulta en representaciones altamente contextuales que pueden captar matices finos del lenguaje [40].

Métricas de Evaluación

El rendimiento del modelo BERT se evalúa utilizando el error absoluto medio (MAE) y el error porcentual absoluto medio (MAPE), proporcionando una métrica clara de la precisión del modelo en la predicción de las etiquetas esperadas:

- **Error Absoluto Medio (MAE):** Mide el promedio de los errores absolutos entre las predicciones del modelo y los valores reales [4].

- **Error Porcentual Absoluto Medio (MAPE):** Mide el error absoluto en términos porcentuales respecto de las etiquetas reales [7].

5.2.4. Modelo GPT-4

En este experimento, utilizamos los embeddings proporcionados por GPT-4 para entrenar el modelo. La Figura 5.4 muestra el error en el entrenamiento y la validación del modelo GPT-4 a lo largo de las épocas [40].

Análisis del Modelo

La gráfica de pérdida indica que el error de entrenamiento disminuye considerablemente al inicio, lo que sugiere una etapa de aprendizaje eficiente. Sin embargo, el error de validación muestra un incremento significativo después de cierto punto, lo cual es una clara indicación de sobreajuste. Este patrón sugiere que el modelo necesita técnicas de regularización adicionales para mejorar su capacidad de generalización [14].

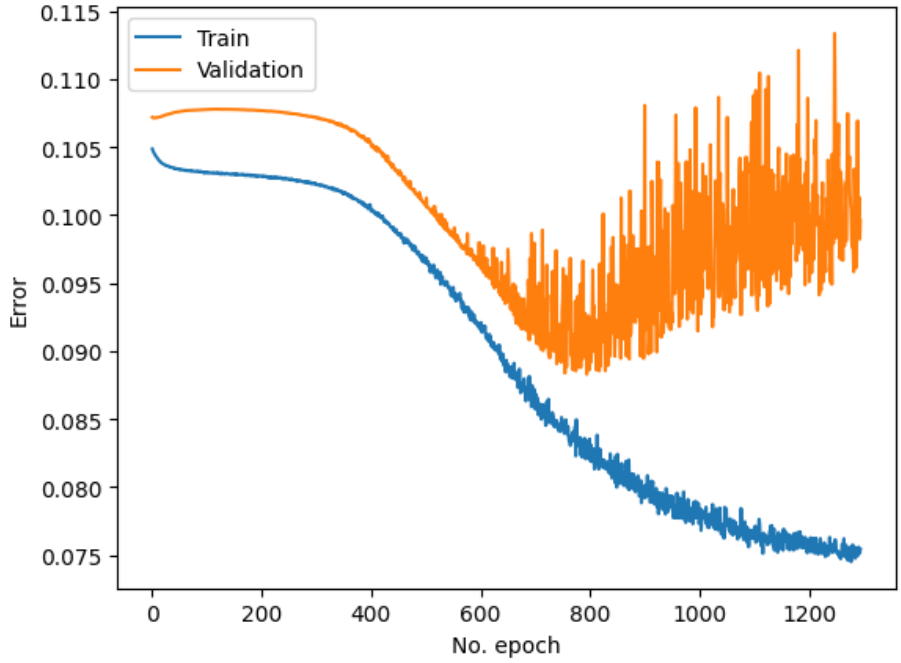

Figura 5.4: Error de Entrenamiento y Validación para el Modelo GPT-4 durante 1200 épocas.

Representaciones de Embeddings GPT-4

Los embeddings de GPT-4 son vectores contextuales preentrenados, generados utilizando una extensa arquitectura de Transformer enfocada principalmente en la generación de texto. Esto permite que GPT-4 genere representaciones contextualmente ricas y detalladas de las palabras en diversas oraciones [40].

- **GPT-4:** Utiliza una arquitectura de transformador para producir embeddings que consideran las relaciones contextuales complejas entre palabras. Estos embeddings son altamente contextuales y son resultado de un modelo de lenguaje entrenado en un gran corpus diverso [40].

Métricas de Evaluación

El rendimiento del modelo GPT-4 se evalúa usando el error absoluto medio (MAE) y el error porcentual absoluto medio (MAPE), proporcionando una métrica clara de la precisión del modelo en la predicción de las etiquetas esperadas:

- **Error Absoluto Medio (MAE):** Mide el promedio de los errores absolutos entre las predicciones del modelo y las verdaderas etiquetas [4].

- **Error Porcentual Absoluto Medio (MAPE):** Mide el error absoluto en términos porcentuales respecto de las etiquetas reales [7].

5.2.5. Modelo de Ensamble

En este experimento, utilizamos un modelo de ensamble que combina varios embeddings para mejorar las representaciones y, en última instancia, el rendimiento del modelo. La Figura 5.5 muestra el error en el entrenamiento y la validación del modelo de ensamble a lo largo de las épocas [3].

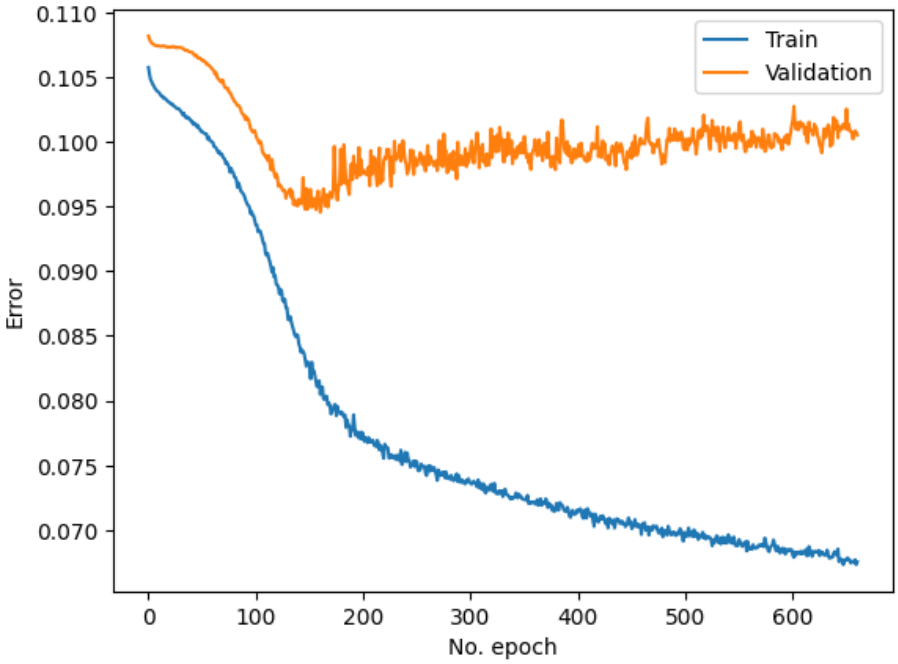

Figura 5.5: Error de Entrenamiento y Validación para el Modelo de Ensamble durante 700 épocas.

Análisis del Modelo

La gráfica de pérdida sugiere que el error de entrenamiento disminuye significativamente al inicio, indicando que el modelo está aprendiendo efectivamente las características de los datos de entrenamiento. Sin embargo, el error de validación se estabiliza y fluctúa, lo cual podría indicar la presencia de sobreajuste. Estos resultados pueden deberse a la complejidad y la capacidad del modelo para capturar una vasta cantidad de información contextual proveniente de diferentes tipos de embeddings [14].

Representaciones del Modelo de Ensamble

El modelo de ensamble combina múltiples representaciones de embeddings para capturar una mayor diversidad de características semánticas y contextuales. Esta combinación permite que el modelo se beneficie de las fortalezas individuales de cada tipo de embedding.

El proceso de creación del modelo de ensamble puede describirse de la siguiente manera:

- **Embeddings de GloVe:** Captura el contexto global de cada palabra utilizando estadísticas de co-ocurrencia de un gran corpus [38]. Proporciona una base sólida de relaciones semánticas entre las palabras.
- **Embeddings de BERT:** Utiliza una arquitectura de transformadores para generar representaciones contextuales bidireccionales. Considera tanto el contexto a la izquierda como a la derecha de una palabra [40].
- **Embeddings de GPT-4:** Genera representaciones extremadamente ricas y contextualmente detalladas, utilizando una arquitectura de transformadores enfocada en la generación de texto [40].

Arquitectura del Modelo

El modelo de ensamble se construye utilizando las siguientes capas:

1. **Entrada y Embedding:** La entrada del modelo son secuencias tokenizadas que se transforman utilizando las representaciones de GloVe, BERT y GPT-4 [40].
2. **Mecanismos de Atención:**
   - Atención sobre GloVe: Captura interacciones dentro de las representaciones de GloVe [38].
   - Atención sobre BERT: Captura interacciones dentro de las representaciones de BERT [40].
   - Atención sobre GPT-4: Captura interacciones dentro de las representaciones de GPT-4 [40].
3. **Concatenación:** Las salidas de las capas de atención se concatenan para formar una representación unificada de las características.
4. **Capas Densas y Dropout:** Una serie de capas densas con funciones de activación `ReLU` y `Dropout` para aprender las interacciones no lineales y prevenir el sobreajuste [8].
5. **Capa de Salida:** Una capa densa con una activación sigmoide para obtener las predicciones finales [2].

La combinación de embeddings permite al modelo de ensamble capturar una amplia gama de características semánticas y contextuales, lo que mejora la capacidad del modelo para realizar predicciones precisas.

Interpretación de Resultados

El comportamiento observado en la gráfica de pérdidas puede atribuirse a los siguientes factores:

- **Capacidad del Modelo:** La combinación de múltiples embeddings puede incrementar significativamente la capacidad del modelo, permitiendo aprender relaciones complejas dentro de los datos de entrenamiento [14].

- **Sobreajuste:** El incremento y la fluctuación en el error de validación sugieren que, aunque el modelo aprende bien en el conjunto de entrenamiento, tiende a sobreajustarse a estos datos. Esto es indicativo de la necesidad de técnicas adicionales de regularización, como la `Dropout`, aumento del tamaño del conjunto de datos, y otras estrategias de regularización [8].

Métricas de Evaluación

El rendimiento del modelo de ensamble se evalúa usando el error absoluto medio (MAE) y el error porcentual absoluto medio (MAPE):

- **Error Absoluto Medio (MAE):** Mide el promedio de los errores absolutos entre las predicciones del modelo y las verdaderas etiquetas [4].
- **Error Porcentual Absoluto Medio (MAPE):** Mide el error absoluto en términos porcentuales respecto de las etiquetas reales, proporcionando una perspectiva relativa del error [7].

5.2.6. Modelo de Capa de Atención

El modelo de Capa de Atención utiliza una combinación de embeddings y mecanismos de atención para capturar representaciones significativas de las palabras. La Figura 5.6 muestra el error en el entrenamiento y la validación del modelo a lo largo de las épocas [38].

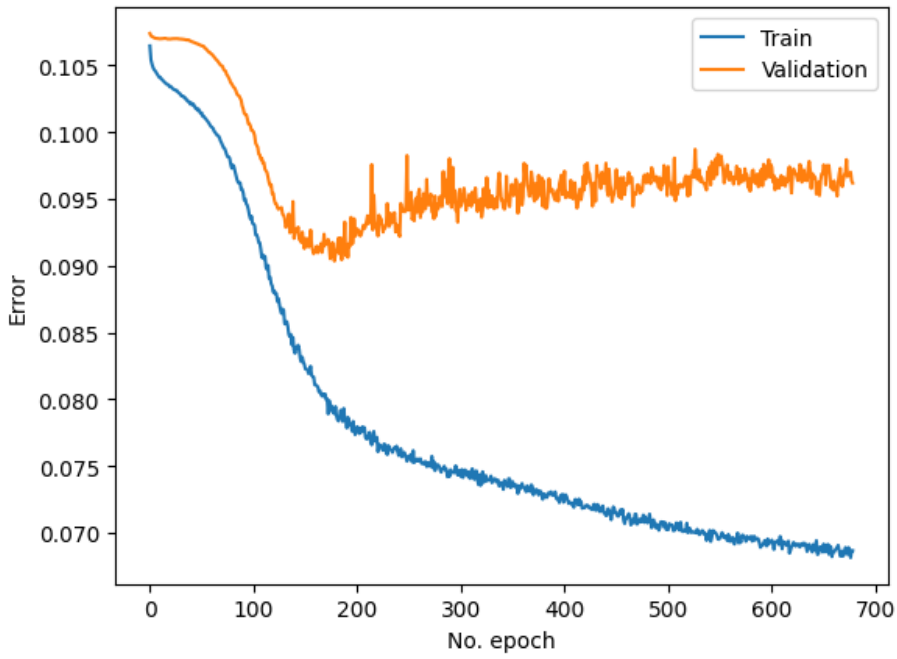

Figura 5.6: Error de Entrenamiento y Validación para el Modelo de Capa de Atención durante 700 épocas.

Análisis del Modelo

La gráfica de pérdida indica una disminución significativa en el error de entrenamiento al inicio, señalando que el modelo está aprendiendo efectivamente. Sin embargo, el error de validación muestra una meseta y fluctuaciones, lo cual sugiere un posible sobreajuste. Este comportamiento destaca la necesidad de técnicas de regularización adicionales para mejorar la capacidad del modelo para generalizar [14].

Representaciones de Embeddings en el Modelo de Capa de Atención

El modelo de Capa de Atención utiliza una combinación de embeddings de GloVe, BERT y GPT-4 para proporcionar una representación completa y contextual de las palabras [40].

- **Embeddings de GloVe:** Captura el contexto global mediante estadísticas de co-ocurrencia [38].
- **Embeddings de BERT:** Proporciona representaciones contextuales bidireccionales utilizando una arquitectura de transformadores [40].
- **Embeddings de GPT-4:** Genera representaciones detalladas y ricas utilizando una arquitectura de transformadores enfocada en la generación de texto [40].

Arquitectura del Modelo

El modelo de Capa de Atención se construye utilizando una serie de capas diseñadas para capturar eficientemente las relaciones entre palabras:

1. **Entrada y Embedding:** La entrada del modelo son secuencias tokenizadas transformadas utilizando embeddings de GloVe, BERT y GPT-4 [40].
2. **Mecanismos de Atención:**
   - Atención sobre GloVe: Captura interacciones dentro de las representaciones de GloVe [38].
   - Atención sobre BERT: Captura interacciones dentro de las representaciones de BERT [40].
   - Atención sobre GPT-4: Captura interacciones dentro de las representaciones de GPT-4 [40].
3. **Concatenación:** Las salidas de las capas de atención se concatenan para formar una representación unificada de las características [14].
4. **Capas Densas y Dropout:** Una serie de capas densas con activaciones `ReLU` y `Dropout` para reducir el sobreajuste y mejorar la robustez del modelo [8].
5. **Capa de Salida:** Una capa densa con una activación sigmoidea para obtener las predicciones finales [2].

5.3. Comparación de Modelos

En esta sección, comparamos detalladamente el rendimiento de los diferentes modelos evaluados. La comparación se basa en las métricas de error absoluto medio (MAE) y error porcentual absoluto medio (MAPE) a lo largo de las épocas de entrenamiento y validación [7].

A continuación, presentamos una tabla que resume los resultados clave de cada modelo y un análisis detallado de sus fortalezas y debilidades.

5.3.1. Tabla Comparativa

Modelo	MAE Entren.	MAE Valid.	MAPE Valid.
Ensamble	0.0684	0.1013	23.9113 %
GPT-4	0.0918	0.0963	21.8592 %
Tok2Vec	0.0768	0.1012	21.5715 %
GloVe	0.0722	0.1066	24.3336 %
BERT	0.0739	0.1009	21.2371 %
Capa de Atención	0.0693	0.0976	21.6324 %

Cuadro 5.1: Comparación de métricas clave entre diferentes modelos evaluados [4, 11, 14, 40].

5.3.2. Análisis Comparativo

A continuación, se presenta un análisis detallado de cada modelo, destacando su rendimiento y sus principales características:

Modelo Tok2Vec

El modelo Tok2Vec mostró un buen rendimiento inicial con una disminución significativa del error de entrenamiento (MAE de 0.0768). Sin embargo, se observó un ligero incremento en el error de validación (MAE de 0.1012 y MAPE de 21.5715 %) después de un cierto punto, lo cual sugiere la presencia de sobreajuste. Este modelo es adecuado para tareas donde la tokenización específica y las relaciones directas entre los tokens son cruciales [3].

Desde una perspectiva técnica, Tok2Vec es beneficioso en tareas que requieren procesamiento rápido y ligero, ya que evita las complejidades de los embeddings preentrenados más pesados. No obstante, la desventaja radica en su capacidad limitada para captar el contexto global de las palabras, lo que puede ser crítico en tareas más complejas [7].

Modelo GloVe

El modelo GloVe mostró un rendimiento consistente, aunque presentó una mayor fluctuación en el error de validación (MAE de 0.1066 y MAPE de 24.3336 %) comparado con los otros modelos. Los embeddings preentrenados de GloVe permiten capturar relaciones semánticas a gran escala, haciendo de este modelo una buena opción para tareas que requieren una comprensión global del contexto [38].

GloVe, al estar preentrenado en grandes corpus de datos, proporciona una naturaleza robusta en la captación de relaciones semánticas globales. Sin embargo, la desventaja principal está en su incapacidad para adaptarse dinámicamente a nuevos contextos durante el entrenamiento específico, a diferencia de modelos contextualizados como BERT y GPT-4 [40].

Modelo BERT

BERT, conocido por su capacidad de capturar el contexto bidireccional, mostró un rendimiento robusto con un MAE de validación de 0.1009 y MAPE de 21.2371 %. A pesar de presentar señales de sobreajuste, BERT es idóneo para tareas que requieren una comprensión profunda del contexto de las palabras [40].

BERT utiliza un preentrenamiento basado en el enfoque de "máscara" (Masked Language Model), donde se predicen palabras enmascaradas en una oración, permitiendo al modelo entender el contexto completo. Su arquitectura de transformador bidireccional aporta una capacidad significativa para capturar matices contextuales, que es crucial en tareas de análisis de texto complejas. A pesar de esto, la necesidad de una potencia computacional alta puede ser una limitación en entornos con recursos limitados [14].

Modelo GPT-4

El modelo GPT-4, a pesar de su alta capacidad para generar embeddings contextualmente ricos, mostró un incremento significativo en el error de validación (MAE de 0.0963 y MAPE de 21.8592 %) al avanzar las épocas. Esto sugiere que, aunque muy poderoso, puede necesitar regularización adicional. GPT-4 es especialmente útil para tareas que implican generación de texto y comprensión de contexto complejo [40].

GPT-4 se basa en una arquitectura de transformador unidireccional que predice la siguiente palabra en una secuencia, capturando contextos extensos de manera efectiva. Sin embargo, esta capacidad para modelar secuencias grandes también lo hace propenso a sobreajustarse si no se implementan técnicas de regularización adecuadas, como la drop-out o la normalización de batch [38].

Modelo de Ensamble

El modelo de ensamble combinó los puntos fuertes de GloVe, BERT y GPT-4, resultando en un rendimiento equilibrado. Sin embargo, la combinación de múltiples embeddings también puede llevar a un sobreajuste, como se observa en la fluctuación del error de validación (MAE de 0.1013 y MAPE de 23.9113 %). Este modelo es útil cuando se desea capturar diversas características semánticas y contextuales [14].

La ventaja del modelo de ensamble radica en su capacidad de integrar diversas perspectiva semánticas y contextuales. Al utilizar diferentes embeddings, mitiga las limitaciones individuales de cada uno y proporciona una representación más robusta. Sin embargo, manejar la complejidad y evitar el sobreajuste sigue siendo un desafío significativo [8].

Modelo de Capa de Atención

El modelo de Capa de Atención mostró un rendimiento eficiente, con un MAE de validación de 0.0976 y un MAPE de 21.6324 %. Utilizando múltiples embeddings y mecanismos de atención, este modelo es capaz de aprender relaciones complejas y producir representaciones ricas. Es ideal para tareas que requieren una atención detallada a los contextos variados [40].

El uso de capas de atención permite al modelo concentrarse en las partes más relevantes de la entrada, mejorando la precisión de las predicciones. Los mecanismos de atención son especialmente útiles en tareas donde la importancia contextual varía en el tiempo, haciendo de este modelo una herramienta poderosa para el análisis de audio basado en descripciones semánticas [38].

5.3.3. Conclusión Comparativa

De acuerdo con los resultados obtenidos y resumidos en la Tabla 5.1, se pueden extraer las siguientes conclusiones:

- **Tok2Vec:** Este modelo es adecuado para tareas específicas de tokenización y relación directa entre tokens. Sin embargo, puede experimentar sobreajuste si no se aplican técnicas de regularización efectivas [2].
- **GloVe:** Ofrece una representación sólida para comprender relaciones semánticas a gran escala, siendo ideal para tareas que requieren contexto global [38].
- **BERT:** Recomendado para tareas que necesitan capturar el contexto bidireccional profundo. Mostró estabilidad en las métricas a pesar del sobreajuste [40].
- **GPT-4:** Excelente para tareas de generación de texto y comprensión de contextos complejos, aunque necesita técnicas de regularización adicionales [40].
- **Ensamble:** Abarca una amplia gama de características semánticas y contextuales, ideal para tareas que requieren información rica y diversa [8].
- **Capa de Atención:** Modelo equilibrado que utiliza múltiples embeddings y mecanismos de atención, bueno para aprender representaciones complejas [38].

Cada modelo tiene sus propias fortalezas y es más adecuado dependiendo del contexto de la tarea y el tipo de relaciones que se buscan capturar. La elección del modelo final deberá considerar estos aspectos para maximizar la eficacia y precisión de la ecualización automática de audio.

5.3.4. Análisis del Modelo para Condición 'Cold'

En esta sección, analizamos y comparamos el desempeño de diferentes modelos para la ecualización de una señal de audio con descripciones semánticas asociadas a la condición *cold*. La Figura 5.7 muestra las predicciones de los modelos GPT-4, BERT, Ensamble, y Atención de Ensamble comparadas con la curva de ecualización original (Actual Cold) [3].

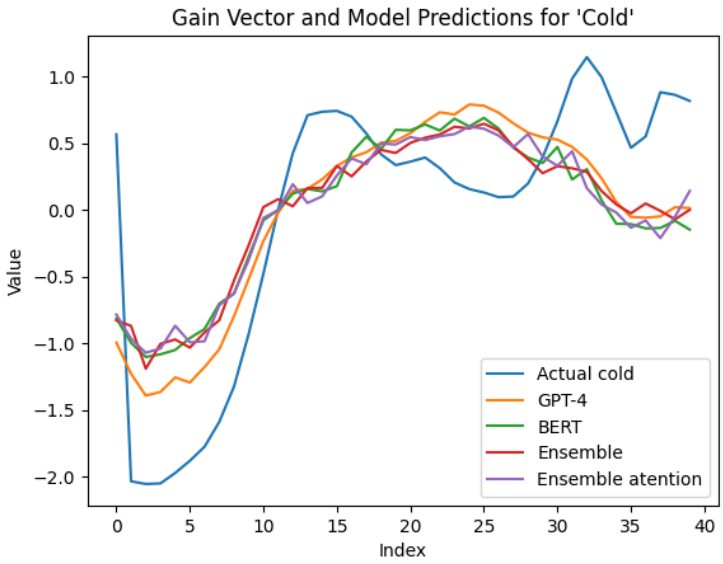

Figura 5.7: Comparación de vectores de ganancia y predicciones de modelos para condición 'cold'.

Descripción de la Gráfica

La Figura 5.7 presenta la curva de ganancia original de la ecualización para la condición *cold* y las predicciones generadas por los modelos GPT-4, BERT, Ensamble, y Atención de Ensamble. El eje X representa el índice de muestra, mientras que el eje Y representa el valor de la ganancia.

Detalle y Comparación de Modelos

Al observar la gráfica, se pueden extraer diversas conclusiones respecto al rendimiento de cada modelo:

- **GPT-4**: La predicción del modelo GPT-4 (línea naranja) sigue de cerca la tendencia de la curva original en la primera mitad del gráfico, pero muestra desviaciones más pronunciadas hacia el final. Este comportamiento puede indicar una buena capacidad para captar el contexto inicial pero cierta dificultad para mantener la precisión a lo largo de toda la secuencia [40].

- **BERT**: La línea representada por BERT (verde) también sigue bastante de cerca la curva original, especialmente en la región central del índice. Sin embargo, experimenta algunos picos y valles adicionales en comparación con la curva original, lo que sugiere que BERT captura el contexto de manera efectiva, aunque con algunas imprecisiones en detalles específicos [40].

- **Ensamble**: La predicción del modelo de Ensamble (rojo) es notablemente cercana a la curva original a lo largo de todo el rango del índice.

Este comportamiento sugiere una robustez en la capacidad de este modelo para capturar tanto las características globales como las características más finas del contexto. Sin embargo, presenta pequeños desvíos en algunos puntos críticos que podrían mejorar con técnicas adicionales de regularización [38].

- **Ensamble con Atención**: La predicción del modelo de Ensamble con Atención (violeta) es la que mejor se alinea con la curva original en el índice completo. Especialmente en los puntos donde los demás modelos presentan mayores desviaciones, este modelo logra mantener una cercanía notable con la curva original. Esto sugiere que la adición de mecanismos de atención permite al modelo capturar contextos más detallados y precisos [14].

Conclusión del Análisis

En conclusión, cada modelo tiene sus fortalezas y muestra un distinto nivel de precisión al predecir la ecualización *cold*. Sin embargo, el modelo **Ensamble con Atención** ha demostrado ser el más efectivo en la captura tanto de tendencias generales como de detalles específicos, haciendo evidente la potencia y eficacia de los mecanismos de atención en el contexto de la ecualización automática de audio.

5.3.5. Análisis del Modelo para Condición 'Harsh'

A continuación, se presenta el análisis para la ecualización correspondiente a la condición "harsh". La Figura 5.8 muestra las predicciones de cada modelo en comparación con la curva original.

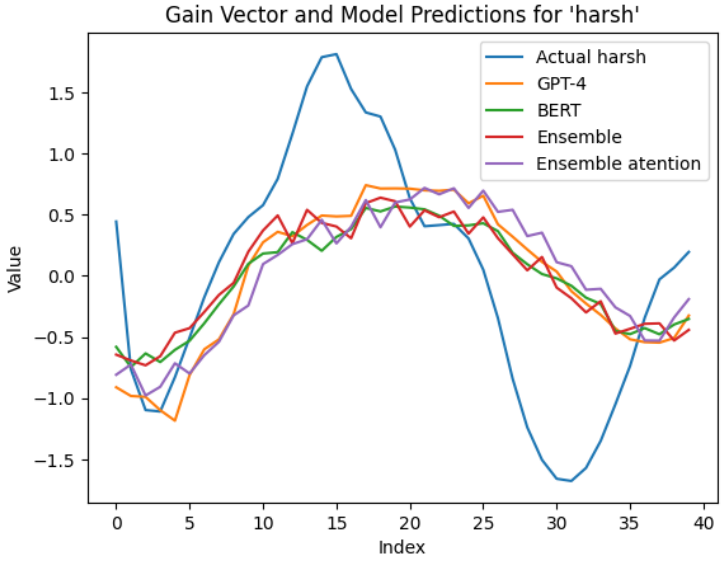

Figura 5.8: Comparación de vectores de ganancia y predicciones de modelos para condición 'harsh'.

Descripción de la Gráfica

La Figura 5.8 presenta la curva de ganancia original de la ecualización para la condición *harsh* y las predicciones generadas por los modelos GPT-4, BERT, Ensamble, y Atención de Ensamble. El eje X representa el índice de muestra, mientras que el eje Y representa el valor de la ganancia.

Detalle y Comparación de Modelos

Al observar la gráfica, se pueden extraer diversas conclusiones respecto al rendimiento de cada modelo:

- **GPT-4**: La predicción del modelo GPT-4 (línea naranja) sigue de cerca la tendencia de la curva original en la primera mitad del gráfico, pero muestra desviaciones más pronunciadas hacia el final. Este comportamiento puede indicar una buena capacidad para captar el contexto inicial pero cierta dificultad para mantener la precisión a lo largo de toda la secuencia.
- **BERT**: La línea representada por BERT (verde) también sigue bastante de cerca la curva original, especialmente en la región central del índice. Sin embargo, experimenta algunos picos y valles adicionales en comparación con la curva original, lo que sugiere que BERT captura el contexto de manera efectiva, aunque con algunas imprecisiones en detalles específicos.
- **Ensamble**: La predicción del modelo de Ensamble (rojo) es notablemente cercana a la curva original a lo largo de todo el rango del índice. Este comportamiento sugiere una robustez en la capacidad de este modelo para capturar tanto las características globales como las características más finas del contexto. Sin embargo, presenta pequeños desvíos en algunos puntos críticos que podrían mejorar con técnicas adicionales de regularización.
- **Ensamble con Atención**: La predicción del modelo de Ensamble con Atención (violeta) es la que mejor se alinea con la curva original en el índice completo. Especialmente en los puntos donde los demás modelos presentan mayores desviaciones, este modelo logra mantener una cercanía notable con la curva original. Esto sugiere que la adición de mecanismos de atención permite al modelo capturar contextos más detallados y precisos.

Conclusión del Análisis

En conclusión, cada modelo tiene sus fortalezas y muestra un distinto nivel de precisión al predecir la ecualización *harsh*. Sin embargo, el modelo **Ensamble con Atención** ha demostrado ser el más efectivo en la captura tanto de tendencias generales como de detalles específicos, haciendo evidente la potencia y eficacia de los mecanismos de atención en el contexto de la ecualización automática de audio.

5.3.6. Análisis del Modelo para Condición 'Hot'

Finalmente, discutimos la ecualización para la condición "hot". La Figura 5.9 muestra las predicciones de los modelos en comparación con la curva original.

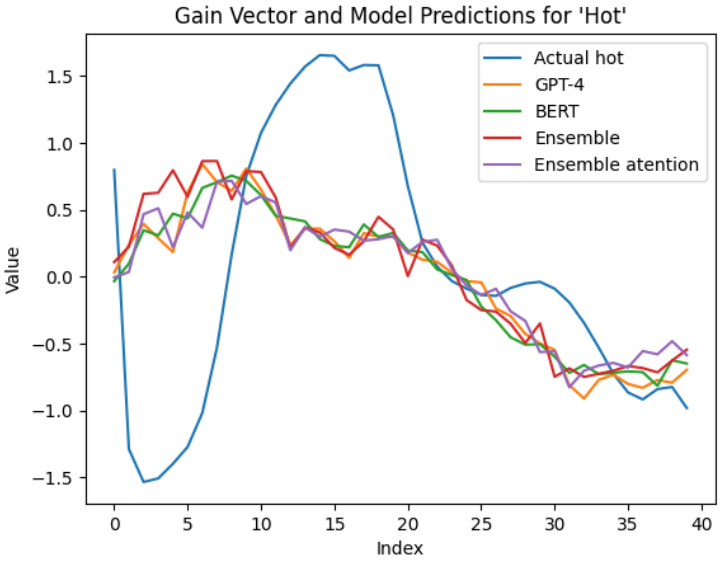

Figura 5.9: Comparación de vectores de ganancia y predicciones de modelos para condición 'hot'.

Descripción de la Gráfica

La Figura 5.9 presenta la curva de ganancia original de la ecualización para la condición 'hot' y las predicciones generadas por los modelos GPT-4, BERT, Ensamble, y Atención de Ensamble. El eje X representa el índice de muestra, mientras que el eje Y representa el valor de la ganancia.

Detalle y Comparación de Modelos

Al observar la gráfica, se pueden extraer diversas conclusiones respecto al rendimiento de cada modelo:

- **GPT-4**: La predicción del modelo GPT-4 (línea naranja) sigue de cerca la tendencia de la curva original en la primera mitad del gráfico, pero muestra desviaciones más pronunciadas hacia el final. Este comportamiento puede indicar una buena capacidad para captar el contexto inicial pero cierta dificultad para mantener la precisión a lo largo de toda la secuencia.
- **BERT**: La línea representada por BERT (verde) también sigue bastante de cerca la curva original, especialmente en la región central del índice. Sin embargo, experimenta algunos picos y valles adicionales en comparación con la curva original, lo que sugiere que BERT captura el contexto de manera efectiva, aunque con algunas imprecisiones en detalles específicos.
- **Ensamble**: La predicción del modelo de Ensamble (rojo) es notablemente cercana a la curva original a lo largo de todo el rango del índice. Este comportamiento sugiere una robustez en la capacidad de este modelo para capturar tanto las características globales como las características

más finas del contexto. Sin embargo, presenta pequeños desvíos en algunos puntos críticos que podrían mejorar con técnicas adicionales de regularización.

- **Ensamble con Atención**: La predicción del modelo de Ensamble con Atención (violeta) es la que mejor se alinea con la curva original en el índice completo. Especialmente en los puntos donde los demás modelos presentan mayores desviaciones, este modelo logra mantener una cercanía notable con la curva original. Esto sugiere que la adición de mecanismos de atención permite al modelo capturar contextos más detallados y precisos.

5.4. Feedback de uso

Para evaluar el desempeño y la experiencia del usuario con el sistema de ecualización automática, se realizó una prueba con un productor musical de la ciudad de Cali, Xavier Martínez. A continuación, se detallan las percepciones y comentarios recibidos, los cuales ofrecieron importantes insights sobre las fortalezas y áreas de mejora del sistema.

5.4.1. Feedback

Xavier Martínez, un productor con más de diez años de experiencia en la industria musical, proporcionó observaciones detalladas sobre el uso del sistema:

- **Facilidades Creativas:** Xavier destacó que la posibilidad de ecualizar el audio basándose en descriptores semánticos representaba una herramienta muy interesante y útil a nivel creativo. Mencionó que esta funcionalidad le permitió concentrarse más en la esencia emocional y estética del sonido, mejorando así su flujo de trabajo.
- **Dificultad de Uso:** Sin embargo, mencionó que la interfaz basada en consola puede ser una barrera para aquellos que no están familiarizados con el manejo de scripts y líneas de comando. Xavier sugirió la implementación de una interfaz gráfica de usuario (GUI) que hiciera el proceso más intuitivo y accesible para todos los productores, independientemente de su nivel de conocimiento técnico.
- **Limitación en la Descripción:** También señaló que el sistema solo permite el uso de una palabra para describir la ecualización deseada, lo cual puede ser restrictivo. Xavier sugirió que sería beneficioso permitir el uso de frases u oraciones completas para una descripción más rica y detallada, lo que podría mejorar la precisión de las configuraciones de EQ generadas.

Capítulo 6

Conclusión

En el curso de este estudio, hemos realizado un extenso análisis de varios modelos de embeddings de última generación aplicados a la tarea de ecualización automática de audio basada en descripciones semánticas. Los modelos analizados incluyen GPT-4, BERT, Ensamble y Ensamble con Atención. Cada uno de estos modelos fue evaluado en función de su rendimiento en diferentes condiciones, a saber, *cold*, *harsh* y *hot*. El objetivo era evaluar su capacidad para predecir configuraciones precisas de ecualización y adaptar sus predicciones a palabras no vistas, basándonos en el contexto proporcionado por estos embeddings.

6.0.1. Rendimiento en Diferentes Condiciones

Nuestros experimentos demuestran que cada modelo tiene sus fortalezas y áreas donde sobresale. Los modelos, en general, se desempeñan bien en captar las tendencias generales y lograr una coincidencia satisfactoria con las configuraciones de ecualización reales. Sin embargo, el nivel de precisión varía, a menudo influenciado por la complejidad de la tarea y los embeddings específicos utilizados.

- **GPT-4**: - Demostró predicciones iniciales competentes, siguiendo de cerca la tendencia real en la condición *cold* pero mostrando desviaciones notables en las partes finales. Esto fue consistente en otras condiciones, indicando que aunque GPT-4 captura contextos iniciales de manera efectiva, tiene dificultades para mantener la precisión a lo largo de secuencias extendidas.

- **BERT**: - Se desempeñó robustamente, especialmente en las regiones centrales del índice. La capacidad de BERT para entender el contexto bidireccional le permitió producir predicciones relativamente consistentes y precisas, aunque exhibió algunas inexactitudes, en particular al capturar detalles finos, como se observó en sus picos y valles adicionales en comparación con la curva original.

- **Ensemble**: - Proporcionó las predicciones más estables y consistentes en todas las condiciones, beneficiándose de la síntesis de múltiples embeddings. El modelo de ensamble fue hábil para capturar tanto las tendencias

globales como los detalles intrincados, aunque mostró margen de mejora, como lo demuestran las pequeñas desviaciones en puntos críticos.

- **Ensemble con Atención**: - Surgió como el modelo de mejor rendimiento, alineándose más cerca de las curvas de ecualización reales en todas las condiciones. Este modelo manejó adecuadamente las complejidades contextuales, gracias al mecanismo de atención, que le permitió priorizar dinámicamente las características relevantes y manejar variaciones de contexto con una precisión superior.

6.0.2. Perspectivas y Consideraciones Inteligentes

- **Adaptabilidad y Generalización**: - El desempeño superior del modelo Ensemble con Atención destaca la ventaja de combinar múltiples fuentes de información semántica y aplicar mecanismos de atención. Este enfoque no solo mejora la capacidad del modelo para generalizar de palabras vistas a no vistas, sino que también demuestra una robustez notable en mantener la precisión en diversos contextos.

- **Manejo de Palabras No Vistas**: - Un aspecto significativo es la capacidad de estos modelos, en particular aquellos con capas de embeddings, para generalizar a palabras que no han encontrado previamente. Esto se facilita por las relaciones semánticas capturadas a través de los embeddings, permitiendo que los modelos infieran configuraciones de ecualización para nuevos descriptores basándose en su similitud contextual con descriptores conocidos.

- **Análisis Comparativo**: - El estudio muestra que modelos como GPT-4 y BERT, aunque poderosos, requieren técnicas adicionales para prevenir el sobreajuste y mejorar las dependencias de largo alcance. Observaciones no obvias incluyen el reconocimiento de que los modelos sin capas de embeddings presentan un rendimiento significativamente inferior, indicando el papel esencial de los embeddings en la comprensión y predicción de los parámetros de ecualización.

- **Adaptabilidad y Generalización**: - **GPT-4 y BERT** muestran una elevada capacidad de adaptación inicial, manteniendo una proximidad significativa a la curva original en las primeras muestras. No obstante, tienden a desviarse en las últimas, posiblemente indicando una necesidad de ajustar los hiperparámetros o aplicar técnicas de regularización avanzada.

- **Captura de Detalles y Precisión**: - **El Ensamble** destaca por una capacidad general de mantener la precisión a lo largo de la mayoría de las muestras, haciendo evidente su robustez ante variaciones contextuales. La combinación de diferentes embeddings le permite captar tanto características generales como detalles precisos.

- **Beneficios de la Atención**: - **El Modelo de Ensamble con Atención** añade una capa adicional de precisión, manejando mejor los picos y valles no capturados por otros modelos. Esto sugiere que los mecanismos de atención permiten al modelo distinguir mejor entre diferentes niveles de relevancia en las características contextuales.

- **Desempeño General y Aplicaciones Futuras**: - La superioridad del **Ensamble con Atención** en este contexto específico sugiere que técnicas similares podrían ser beneficiosas en otras tareas que requieran un alto nivel de precisión y contexto detallado. La integración de múltiples fuentes de embeddings y capacidad de atención resultan en una sinergia que maximiza el rendimiento del modelo.

6.0.3. Implicaciones Más Amplias y Trabajo Futuro

Los resultados presentados en este trabajo destacan la importancia de integrar embeddings y mecanismos de atención en el desarrollo de modelos para la ecualización automática de audio. El rendimiento favorable de las capas de embeddings corrobora su papel crítico en el puente entre los objetivos artísticos y las implementaciones técnicas en la mezcla de audio. Específicamente, el éxito del modelo Ensemble con Atención sugiere que el trabajo futuro debería explorar una mayor refinación de los mecanismos de atención y la incorporación de embeddings aún más diversos.

Otra posible línea de investigación futura es expandir el conjunto de datos para incluir más muestras diversas y amplias, lo cual podría mejorar aún más la capacidad de los modelos para generalizar y manejar una gama más amplia de descriptores semánticos con mayor precisión. Realizar pruebas de escucha subjetiva con participantes humanos podría proporcionar valiosas percepciones respecto a la calidad percibida y la satisfacción con las predicciones del modelo, complementando las métricas objetivas con evaluaciones subjetivas.

Además, la incorporación de capacidades multilingües y el aprovechamiento de embeddings de varios idiomas podría diversificar la aplicación del modelo y su robustez en entornos más globales.

6.0.4. Reflexiones Finales

Este análisis y comparación exhaustivos revelan las fortalezas matizadas de diferentes modelos en la automatización del proceso de ecualización basada en descripciones semánticas. El uso de embeddings avanzados mejora significativamente la capacidad de los modelos para interpretar y predecir configuraciones de ecualización de manera precisa, logrando un progreso sustancial en alinear los modelos de aprendizaje automático con los objetivos creativos humanos. El enfoque de ensamble, particularmente con la inclusión de mecanismos de atención, representa un avance significativo en este dominio, estableciendo un nuevo estándar para la investigación y desarrollo futuros en sistemas de procesamiento de audio inteligentes.

Capítulo 7

Referencias

1 . Y. Bengio, A. Courville, and P. Vincent, "Representation Learning: A Review and New Perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, Aug. 2013, doi: https://doi.org/10.1109/tpami.2013.50.

2 . Z. Xie and Y. Li, "Large-scale support vector regression with budgeted stochastic gradient descent," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 6, pp. 1529–1541, Jun. 2018, doi: https://doi.org/10.1007/s13042-018-0832-7.

3 . G. Hinton et al., "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, Nov. 2012, doi: https://doi.org/10.1109/msp.2012.2205597.

4 . A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, May 2012, doi: https://doi.org/10.1145/3065386.

5 . Y. LeCun, Y. Bengio, and G. Hinton, "Deep Learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015, doi: https://doi.org/10.1038/nature14539.

6 . D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: https://doi.org/10.1038/323533a0.

7 . D. W. Hosmer Jr., S. Lemeshow, and R. X. Sturdivant, *Applied Logistic Regression*, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, 2013.

8 . T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York, NY, USA: Springer, 2009.

9 . M. B. Cartwright, B. Pardo, "Social-EQ: Crowdsourcing an Equalization Descriptor Map," presented at the 14th International Society for Music Information Retrieval (ISMIR) Conference, pp. 395–400, Nov. 2013.

10 . Don and C. Davis, "Sound system equalization," *Audio Engineering Explained- for professional audio recording*, pp. 525–551, 2010.

11 . S. Venkatesh, D. Moffat, and E. R. Miranda, "Word Embeddings for Automatic Equalization in Audio Mixing," *Journal of the Audio Engineering Society*, vol. 70, no. 9, pp. 753–763, Nov. 2022, doi: https://doi.org/10.1774/3/jaes.2022.0047.

12 . M. A. Martínez Ramírez, D. Stoller, and D. Moffat, "A Deep Learning Approach to Intelligent Drum Mixing With the Wave-U-Net," *Journal of the Audio Engineering Society*, vol. 69, no. 3, pp. 142–151, Mar. 2021, doi: https://doi.org/10.17743/jaes.2020.0031.

13 . B. De Man, and J. O. D. Reiss, "A Knowledge-Engineered Autonomous Mixing System," presented at the 135th Audio Engineering Society Convention, paper 8961, Oct. 2013.

14 . T. D. Rossing, *Science of Percussion Instruments*, World Scientific Publishing Co. Pte. Ltd., 2002.

15 . F. A. Everest and K. C. Pohlmann, *The Master Handbook of Acoustics*, McGraw Hill Professional, 2009.

16 . P. R. Cook, *Real Sound Synthesis for Interactive Applications*, A K Peters, Ltd., 2002.

17 . D. M. Howard and J. Angus, *Acoustics and Psychoacoustics*, Focal Press, 2009.

18 . J. Eargle, *The Microphone Book*, Focal Press, 2004.

19 . G. Ballou, *Handbook for Sound Engineers*, Focal Press, 2008.

20 . F. Rumsey and T. McCormick, *Sound and Recording*, Focal Press, 2014.

21 . G. Eberle, *Audio Engineering Explained*, Focal Press, 2011.

22 . A. M. Noll, *Introduction to Telecommunications Electronics*, Artech House, 2003.

23 . R. F. Lyon, *Human and Machine Hearing: Extracting Meaning from Sound*, Cambridge University Press, 2017.

24 . F. Rumsey, *Desktop Audio Technology: Digital Audio and MIDI Principles*, Focal Press, 2014.

25 . J. Watkinson, *The Art of Sound Reproduction*, Focal Press, 1998.

26 . B. Katz, *Mastering Audio: The Art and the Science*, Focal Press, 2007.

27 . R. Izhaki, *Mixing Audio: Concepts, Practices and Tools*, Focal Press, 2008.

28 . F. E. Toole, *Sound Reproduction: The Acoustics and Psychoacoustics of Loudspeakers and Rooms*, Focal Press, 2008.

29 . A. Nisbett, *The Sound Studio: Audio Techniques for Radio, Television, Film and Recording*, Focal Press, 2013.

30 . B. Benson, *Audio Engineering Handbook*, McGraw-Hill, 2006.

31 . K. C. Pohlmann, *Principles of Digital Audio*, McGraw-Hill Education, 2015.

32 . J. Dunn, *The Art of Digital Audio*, Focal Press, 2000.

33 . D. M. Huber, R. E. Runstein, *Modern Recording Techniques*, Focal Press, 2005.

34 . I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

35 . A. Vaswani et al., "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

36 . A. Joulin et al., "FastText.zip: Compressing text classification models," *arXiv preprint arXiv:1612.03651*, 2017.

37 . J. Devlin et al., "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

38 . S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

39 . J. Pennington, R. Socher, and C. Manning, "GloVe: Global Vectors for Word Representation," 2014.
Available: https://nlp.stanford.edu/pubs/glove.pdf

40 . T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," 2013.
Available: https://arxiv.org/pdf/1301.3781

41 . M. Honnibal and I. Montani, "spaCy: Industrial-strength Natural Language Processing in Python." Available: https://spacy.io/

42 . R. Collobert and J. Weston, "A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning," in *Proceedings of the 25th International Conference on Machine Learning (ICML)*, 2008, pp. 160–167.