Síntesis de espectrogramas de sonidos subacuáticos con Autoencoders y Transfer Learning

María C. Cebedio y Marco Carnaghi
ICYTE, Depto. de Electrónica y Computación, Facultad de Ingeniería - UNMDP
Mar del Plata, 7600, Argentina
{mcarnaghi,celestecebedio}@fi.mdp.edu.ar

***Resumen*—En este trabajo se presenta la síntesis de espectrogramas de sonidos subacuáticos de baja frecuencia, a partir de aplicar el método de aprendizaje por transferencia (Transfer Learning) en Autoencoder Convolucionales simples.**

El estudio abarca el acondicionamiento de los datos, la utilización de métodos de Transfer Learning con modelos pre-entrenados sencillos, el análisis de la mejor opción y la generación de los espectrogramas ficticios. La síntesis final se realiza a partir de vectores de baja dimensionalidad generados con distribución gaussiana.

Los resultados obtenidos demuestran que, a partir de pocos datos de muestra, con arquitectura de baja complejidad y asumiendo una distribución normal de los vectores reales, es posible sintetizar espectrogramas de sonidos de baja frecuencia subacuáticos.

***Palabras Claves*—Autoencoders convolucionales, espectrogramas, Tranfer Learning, sonidos subcuáticos, síntesis.**

I. Introducción

Las características del medio submarino hacen ideal la transmisión acústica y, por este motivo, el estudio del entorno y todo lo concerniente a él se basa en señales de audio. Una gran cantidad de trabajos en el área requieren acceso a registros reales de sonidos subacuáticos para sus análisis, pero las pruebas en campo son costosas, requieren de infraestructura, suelen llevar un tiempo considerable y, en muchos casos, dependen de factores externos [1]. Por estos motivos, la ausencia del volumen requerido de datos de audio submarino es un problema habitual.

En consecuencia, la posibilidad de generar datos artificialmente podría ser de utilidad para los primeros pasos de una investigación, entrenamiento de redes neuronales, simulación de entornos marinos, etc. Asimismo, si estos datos ficticios pueden ser generados en tiempo real por dispositivos de baja potencia computacional, podrían generarse bancos de prueba artificiales, lo cual sería muy beneficioso para el estudio del medio.

En la actualidad se han diseñado modelos de aprendizaje profundo (ML) generativos basados en técnicas de aprendizaje profundo no supervisado, como el Autoencoder Convolucional (AE-CNN) y el Autoencoder Variacional (VAE) para reconstruir señales [2]. Sin embargo, estos modelos requieren disponer de grandes volúmenes de datos, amplia capacidad de cómputo y largos períodos de tiempo de entrenamiento. Debido a esto, el Transfer Learning (TL) ha surgido como una técnica atractiva y se emplea con el objetivo de reutilizar un modelo previamente entrenado, como punto de partida en una nueva tarea o aplicación [3]. La aplicación de metodologías de TL incrementa el grado de generalización de los modelos de ML, permitiendo la utilización de conjuntos de datos de menor tamaño y mayor simplicidad (baja dimensionalidad). Además, posibilita una aceleración en el entrenamiento, permite representaciones más robustas y de amplia aplicación [4].

Existen modelos pre-entrenados, que han demostrado ser útiles para realizar TL en la etapa de extracción de características. El problema de estos modelos radica en la complejidad de los mismos. Los modelos típicamente utilizados para realizar TL poseen una cantidad de capas profundas alta y , por ende, la cantidad de parámetros asociados también. A modo de ejemplo, en el caso de AlexNet presenta 62,3 millones de parámetros [5], lo que implica igual orden de operaciones aritméticas y necesidad de trabajar en paralelo con GPU's. Por ende, la complejidad asociada a cualquier implementación de este modelo en un sistema embebido es muy grande. De la misma manera Densenet201 [6] presenta 201 capas convolucionales profundas.

En este contexto, el objetivo del presente trabajo radica en la generación de espectrogramas ficticios partiendo de un escaso volumen de datos de entrenamiento y aplicando técnicas de TL [7] con arquitecturas de Autoencoders de baja complejidad a nivel de operaciones y de pocas capas convolucionales. La elección de la arquitectura y la evaluación de un modelo sencillo, es el punto de partida para la posible implementación en un sistemas embebido que posea bajo potencia computacional.

II. Metodología

La metodología de síntesis consiste en: Adecuación de los datos de entrada, Selección de modelos pre-entrenados con set de datos correspondiente a ballenas barbadas, Utilizar técnicas de TL sobre arquitecturas pre-entrenadas, evaluación de resultados y obtención del modelo, Utilizar técnicas de TL a conjuntos de datos minoritarios correspondientes sonidos subacuáticos y generación de espectrogramas ficticios.

II-A. Adecuación de los datos de entrada

El objetivo de esta etapa es obtener una matriz bidimensional que represente el espectrograma de magnitud logarítmica para cada registro de audio. El conjunto de todos los espectrogramas normalizados entre 0 y 1 para ser interpretados como una imagen en escala de grises, se

agrupa en un tensor. Las características de los registros de audio se presentan en [8].

Los datos recolectados corresponden a audios de diferentes sonidos de baja frecuencia presentes en el entorno submarino. De esta manera, se tienen registros agrupados en diferentes clases: sub-especies de Ballenas Barbadas, Peces y Ruidos de mar. En la Tabla I se resumen la cantidad de datos disponibles y la forma del tensor de entrada para cada categoría.

DATOS	N	Dimensión del tensor de entrada	
		Datos Train	Datos Test
B. Barbadas (todas)	6715	[6043,128,196,1]	[672,128,196,1]
B. Azul	163	[146,128,196,1]	[17,128,196,1]
B. Gris	55	[49,128,196,1]	[6,128,196,1]
B. Rorcual	2884	[2595,128,196,1]	[289,128,196,1]
B Groenlandia	434	[393,128,196,1]	[44,128,196,1]
B. Minke	122	[109,128,196,1]	[13,128,196,1]
B. Jorobada	2879	[2591,128,196,1]	[288,128,196,1]
B. Franca	147	[132,128,196,1]	[15,128,196,1]
Peces	313	[281,128,196,1]	[32,128,196,1]
Ruido de Mar	1625	[1462,128,196,1]	[163,128,196,1]

Tabla I: Datos disponibles para el entrenamiento de los modelos: cantidad y forma del tensor de entrada

II-B. Modelos pre-entrenados.

Los modelos de AE-CNN y VAE utilizados durante el pre-entrenamiento cuentan con 4 capas en la etapa de extracción de características. En todas ellas, la función de activación es tipo RELU, el Padding se re-acomoda de forma automática y se utiliza la operación de Batch Normalization [9].

El modelo AE-CNN, tiene una sola capa densa en la sección de capa profunda y para VAE dicha capa está representada por capas densamente conectadas de varianza y media. Estos modelos entrenados y las características del entrenamiento con los datos mayoritarios correspondientes a la categoría “Ballenas Barbadas”se presentan en [8].

La dimensión del espacio latente que presenta mejor performance de cara a su utilización como modelos pre-entrenados surge de una análisis previo realizado en [10]. En dicho análisis se comparan arquitecturas de 1, 2 , 3 y 4 dimensiones para las dos arquitecturas, pero entrenadas con el conjunto de todas las ballenas barbadas, obteniéndose un error de entrenamiento menor con D=4 para AE-CNN y D=3 para VAE.

II-C. Transfer Learning

Esta técnica permite sacar provecho de la experticia ganada en las capas convoluciones de modelos entrenados con una gran cantidad de datos y utilizarlas para extraer características de otras conjunto minoritario de datos.

El concepto principal radica en congelar, es decir, fijar los pesos de ciertas capas durante el entrenamiento y hacer un ajuste fino de los pesos restantes para responder al nuevo problema. Esta estrategia permite reutilizar los conocimientos en términos de la arquitectura global de la red y explotar sus estados como punto de partida para el entrenamiento [11]. En este caso en particular, se congelan los pesos de las capas convolucionales y se re-entrenan las capas densamente conectadas. En la Fig. 1 se muestra el modelo AE-CNN y las capas que se “congelan”, ajustando los pesos únicamente de las capas densamente conectadas.

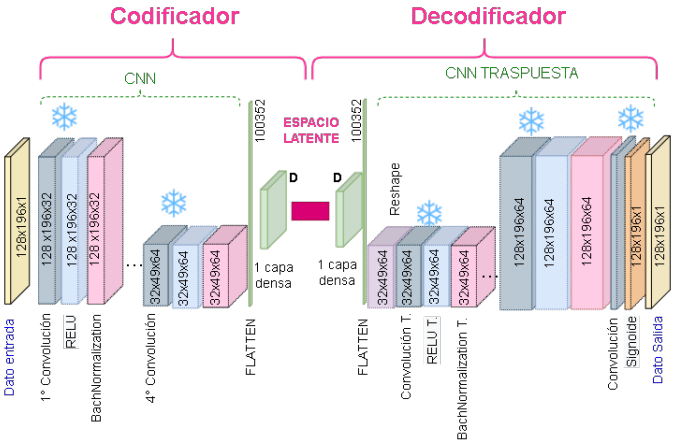

Fig. 1: Arquitectura Autoencoder CNN para TL. Las capas convolucionales marcadas con el símbolos son aquellas que se “congelan”.

III. Análisis de los resultados

El resultado de implementar cada arquitectura, muestra una cantidad de parámetros entrenables y otros fijos, que dan un noción de la dimensión del modelo, la complejidad de las operaciones involucradas y los tiempos de entrenamiento requeridos. Los parámetros resultan de aplicar TL a CNN son: 1.137.093, de los cuales 932.805 son entrenables y los parámetros resultan de aplicar TL a VAE son: 1.209.223, de los cuales 1.004.358 son entrenables.

III-A. Re-entrenamiento para subespecies de ballenas

En este punto, se re-entrenan los modelos seleccionados con distintos sub-clases de ballenas, se calcula el error cuadrático medio (MSE) y el error de similitud estructural (SSIM). Para realizar comparativa de la mejora obtenida al utilizar TL, se entrenan los modelos AE y VAE, sin haber realizado un pre-entrenamiento. A modo de resumen, en la tabla II, se presentan los resultados obtenidos. Del análisis de estos datos presentados, resulta que aplicar TL a la arquitectura AE-CNN pre-entrenada con el conjunto de todas las ballenas Barbadas, posee el mejor desempeño.

Datos de TEST	N	CNN		TL CNN		VAE		TL VAE	
		MSE	SSIM	MSE	SSIM	MSE	SSIM	MSE	SSIM
B. Azul	146	0.026	0.590	0.008	0.64	0.024	0.598	0.053	0.499
B. Minke	109	0.068	0.345	0.009	0.553	0.011	0.519	0.013	0.482
B. Jorobada	2591	0.026	0.392	0.007	0.452	0.052	0.365	0.017	0.397
B. Franca	132	0.066	0.413	0.016	0.478	0.050	0.433	0.017	0.322
B. Rorcual	2595	0.006	0.460	0.006	0.463	0.029	0.324	0.061	0.139
B. Gris	49	0.038	0.412	0.005	0.579	0.059	0.384	0.011	0.49
B. Groenlandia	393	0.024	0.546	0.003	0.629	0.03	0.535	0.018	0.507

Tabla II: Resumen de métricas obtenidas para las diferentes arquitecturas, con y sin Transfer Learning (TL)

En la Fig. 2, se muestra un espectrograma correspondiente a la categoría “Ballena Groenlandia” y las distintas reconstrucciones realizadas. Se observa que la reconstrucción obtenida con la arquitecturas CNN y VAE, sin aplicar TL, son muy desalentadoras. Esto es de esperarse ya que la cantidad de datos de esta subclase es considerablemente reducida. Sin embargo, al realizar TL se aprecia una mejora del desempeño.

III-B. Extensión a otras fuentes de sonidos

El modelo AE-CNN pre-entrenado [8] se utiliza para realizar TL, con el fin de adecuarlo a otros sonidos subacuáticos. En este caso en concreto, la técnica es evaluada en base a sonidos provenientes de peces y ruido de mar.

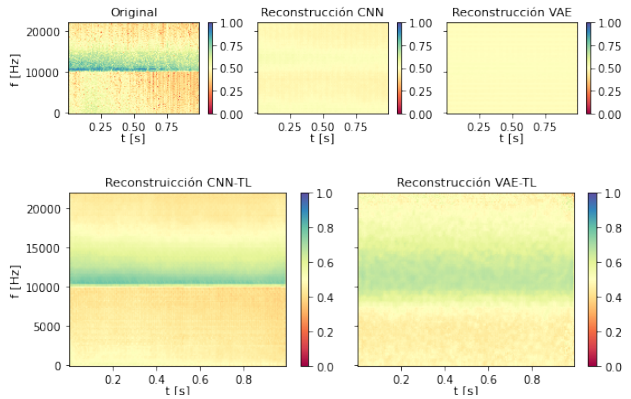

Fig. 2: Espectrograma Ballena Groenlandia y sus diferentes reconstrucciones. La cantidad de datos de entrenamiento es $N_{train} = 393$.

La Fig. 3 muestra las reconstrucciones obtenidas para estos registros. Se observa que el modelo pre-entrenado aporta una etapa de extracción de características de utilidad en ambos casos y las reconstrucciones obtenidas recuperan las propiedades esenciales de ambas fuentes sonoras.

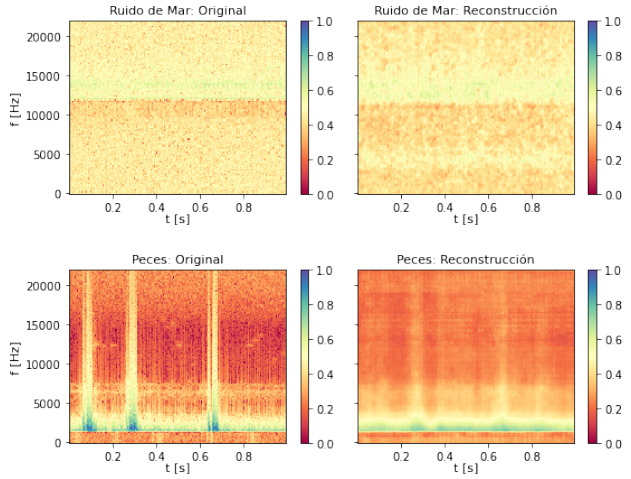

Fig. 3: Espectrogramas de sonidos subacuáticos: Espectrograma Original vs Reconstrucción realizada con TL aplicado a AE-CNN.

IV. Generación de espectrogramas ficticios

Para la generación de datos ficticios se utiliza sólo el bloque decodificador del Autoencoder CNN y se ingresa a éste (según sea el espectrograma a generar), con un vector de dimensión 4.

Se asume que la distribución de cada valor del espacio latente es Gaussiana y, a partir de todos los vectores de espacio latente generados con los datos de Train, se obtiene un valor medio y una desviación. Estos valores actúan como punto de partida para generar vectores de códigos aleatorios de dimensión 4 a ingresar en el bloque decodificador entrenado.

A modo de ejemplo, la Fig. 4 muestra imágenes sintetizadas y la Fig. 5 imágenes reales, para ballenas "Groenlandia".

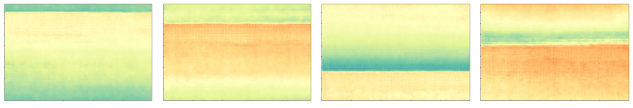

Fig. 4: Espectrogramas sintetizados, obtenidos con un decodificador entrenado, a partir de registros de ballenas Groenlandia.

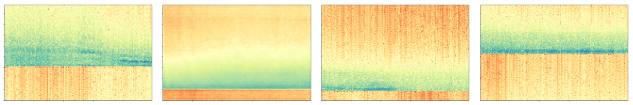

Fig. 5: Espectrogramas Reales, obtenidos a partir de registros aleatorios de ballenas Groenlandia.

V. Conclusión y trabajo a futuro

En este trabajo se logra la generación de espectrogramas ficticios, correspondientes a fuentes sonoras de baja frecuencia del entorno submarino. La propuesta de utilizar técnicas de TL, dado el marcado desbalance en los datos disponibles, muestra buenos resultados.

El modelo presentado posee un reducido número de capas, las operaciones involucradas son simples y la cantidad de parámetros a implementar son bajos, en comparación con las arquitecturas de código abierto usualmente utilizadas. Esta particularidad le permite un entrenamiento en una PC personal y una posible implementación en un sistema embebido de baja potencia computacional, donde las operaciones aritméticas y errores asociados a estas, sumado a la cantidad de parámetros son un punto importante a tener en cuenta. A medida que las redes se vuelven más profundas y complejas, el costo de una implementación aumenta.

Las innovadoras herramientas de código abierto, tales como Tensor Flow Lite [12] o aquellas que permiten la utilización de código python sobre FPGA, facilitan la implementación de este tipo de redes de baja complejidad.

VI. Agradecimientos

Al Dr. Diego Comas y al Dr. Gustavo Meshino por los conocimientos impartidos sobre la temática.

Referencias

[1] E. Tejero, "Aplicaciones de Machine Learning a la Bioacústica Marina," Ph.D. dissertation, 07 2020.

[2] Q. Xu, Z. Wu, Y. Yang, and L. Zhang, "The difference learning of hidden layer between autoencoder and variational autoencoder," in *29th Chinese Control And Decision Conference*, 2017, pp. 4801–4804.

[3] D. Sarkar, R. Bali, and T. Ghosh, *Hands-On Transfer Learning with Python*, 1st ed. Packt, 1995, ISBN 13: 9781788831307.

[4] L. Torrey and J. Shavlik, *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, 1st ed. IGI Global. Hershey, 2010, DOI:10.4018/978-1-60566-766-9.ch011.

[5] J. Wei, "AlexNet: The Architecture that Challenged CNNs," Towards Data Science, 2019. [Online]. Available: https://acortar.link/IrMULc(acceso:25dejuniode2022).

[6] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2261–2269.

[7] E. Tsalera, A. Papadakis, and M. Samarakou, "Comparison of Pre-Trained CNNs for Audio Classification Using Transfer Learning," *Journal of Sensor and Actuator Networks*, vol. 10, p. 72, 12 2021.

[8] M. C. Cebedio and M. Carnaghi, "Repositorio-CASE2022," GitHub, 2022. [Online]. Available: https://github.com/Reposinnombre/CASE2022

[9] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," 2015. [Online]. Available: https://arxiv.org/abs/1502.03167

[10] M. Carnaghi and M. C. Cebedio, "Espectrogramas de registros de Ballenas Barbadas, sintetizados a partir de Autoencoders," *Congreso Argentino de Sistemas Embebidos CASE*, 08 2022.

[11] A. Yoss, "Transfer Learning using Pre-Trained AlexNet Model and Fashion-MNIST," Towards Data Science, 2020. [Online]. Available: https://acortar.link/IBrXLm(acceso:25dejuniode2022).

[12] "Tensorflow lite," TensorFlow. [Online]. Available: https://www.tensorflow.org/lite/guide?hl=es-419(acceso:23dejuliode2022)