

Hacia la implementación *on the edge* de un segmentador de PCG basado en la U-Net

Daniel Enériz¹, Antonio J. Rodríguez-Almeida², Himar Fabelo³,
Nicolás Medrano¹, Belén Calvo¹, Gustavo M. Callico²

¹ Grupo de Diseño Electrónico (GDE)
Instituto de Investigación en Ingeniería de Aragón (I3A)
Universidad de Zaragoza, Mariano Esquillor s/n, 50018, Zaragoza, Spain.
Tel. +34-976762707, e-mail: eneriz@unizar.es

²Instituto Universitario de Microelectrónica Aplicada (IUMA), Universidad de Las Palmas de Gran Canaria (ULPGC)

³Fundación Canaria Instituto de Investigación Sanitaria de Canarias (FIISC), Las Palmas de Gran Canaria

Resumen

Un sistema de asistencia al diagnóstico de enfermedades cardiovasculares requiere de precisión y respuesta en tiempo real, algo que se puede alcanzar gracias a la implementación de modelos *deep learning on the edge*. En este trabajo se presenta la reducción de un modelo para la segmentación de fonocardiogramas y su efecto en la implementación sobre FPGAs *low-spec*.

Introducción

Las enfermedades cardiovasculares (CVD) son la mayor causa de muerte, con un 32 % de fallecimientos. A pesar de que la mayoría de estos casos se podrían detectar con una simple oscultación cardiaca, el 75 % de las muertes se dan en países en vías de desarrollo, donde no existen programas de detección temprana. Por todo ello, es interesante el desarrollo de un dispositivo de ayuda al diagnóstico de CVDs a través del análisis de fonocardiogramas (PCG), que son las grabaciones de los sonidos que produce el corazón durante su actividad mecánica. Dichas grabaciones están compuestas, fundamentalmente, por dos sonidos: *S1* y *S2*; los cuales marcan el inicio de las dos etapas del ciclo cardiaco: *sístole* y *diástole*. La segmentación del PCG en cada uno de los estados cardiacos (Figura 1) es una de las tareas más básicas del procesado automático PCGs. Durante los últimos años se han presentado diferentes trabajos basados en modelos ocultos de semi-Markov (HSMM) [1-3], pero destaca sobre ellos un modelo basado en la arquitectura U-Net [4], una red neuronal convolucional (CNN) con gran éxito en la segmentación de imágenes.

Otra cuestión que se debe tener en cuenta es el soporte físico del dispositivo de ayuda al diagnóstico, ya que debe dar una respuesta en tiempo real. Puesto que gran parte del impacto de las CVDs se produce en países en vías de desarrollo, es deseable que dicho

dispositivo sea capaz de funcionar independientemente de los recursos disponibles. Por ello, un procesado *on the edge* es la mejor opción, evitando conexiones a Internet y pudiendo ser alimentado por batería. Sin embargo, debido a la limitada potencia computacional que ofrecen este tipo de sistemas, es preciso realizar un estudio de los requerimientos computacionales del algoritmo para su optimización en el hardware seleccionado.

En este trabajo se presenta una adaptación del segmentador U-Net [4], que incluye tres parámetros de reducción del tamaño del modelo. Además, se ha explorado el impacto de dichos parámetros en los recursos de dos *field-programmable gate arrays* (FPGA) de bajas prestaciones, usando para ello una implementación de la red con datos de punto fijo de baja resolución.

Metodología

Base de datos y preprocesado

La base de datos del *Physionet/Cinc Challenge* de 2016 [5-6] se ha usado para evaluar los modelos, al igual que se hizo en [4]. Siguiendo el mismo preprocesado, se han extraído cuatro envolventes: el valor absoluto de la transformada de Hilbert, el homomórfico, la densidad de potencia espectral entre 40 y 60 Hz y el nivel d4 de la transformada *wavelet*. Finalmente, dichas envolventes se han decimado a 50 Hz para reducir el impacto computacional.

Arquitectura del modelo

La red es una adaptación de la U-Net para funcionar con señales unidimensionales, como el PCG, con una entrada de N muestras temporales de las envolventes ($N \times 4$). Está compuesta por cuatro *encoders* y cuatro *decoders*. Los primeros están formados por dos capas *Conv1D* con activación *ReLU*, seguidas por un

maxpooling, de forma que se consigue duplicar el número de canales y reducir a la mitad la dimensión temporal. Por su parte, los *decoders*, cuentan con dos entradas: la capa anterior y una *skip connection*. Estos están formados por una primera etapa donde se sobremuestra la información temporal proveniente de la capa anterior, la cual se pasa por una *Conv1D* con activación *ReLU*. Tras ello, la información resultante se concatena con la conexión residual, lo cual va seguido de dos *Conv1D* con activación *ReLU*. Todo se configura de forma que a la salida de cada *decoder* se dobla la dimensión temporal y se reduce a la mitad la de canales. Finalmente, tras el último *decoder* se coloca una *Conv1D* con activación *softmax*, que da una salida $N \times 4$, asociada a la probabilidad de estar en cada uno de los estados cardiacos para cada instante temporal.

Parámetros de reducción

En este trabajo, se han considerado tres parámetros de reducción del modelo: N , n_0 y n_{enc} . El primero, N , se corresponde con el número de muestras a la entrada, considerando los mismos valores que en [4]: 64, 128, 256 y 512. El segundo, n_0 , es el número de canales que se producen en el primer encoder, que luego condiciona el resto de la arquitectura. En [4] está fijo en 8, y en este trabajo se ha reducido hasta 4 en pasos de 1. Por último, n_{enc} es el número de *encoders/decoders*, que es 4 en el trabajo original y el cual se ha reducido hasta 1 en pasos de 1. La combinación de todos los parámetros permite obtener 80 versiones diferentes del modelo, las cuales se han entrenado y validado usando el método presentado en [4]. Los resultados muestran que la reducción de n_{enc} a 3 o 2 escasamente afecta al desempeño del modelo. Lo mismo ocurre con n_0 y N , cuyos efectos son mínimos siempre que $n_{enc} > 1$.

Implementación

Usando una descripción del modelo en C/C++, a través de una herramienta de síntesis (HLS) y usando datos de punto fijo de 16 bits, se ha podido evaluar los efectos de estos parámetros de reducción en los recursos de las FPGAs XC7Z010 y XC7Z020. En ellos se muestran que tanto la *block random-access memory* (BRAM) como las *look-up tables* (LUTs) son los recursos limitantes, a consecuencia de la necesidad de guardar un gran número de parámetros y mapas de características. En concreto, los modelos con $n_{enc} > 2$ requieren más LUTs de las disponibles en la XC7Z010, y la BRAM necesaria en el modelo original ($n_0=8$, $n_{enc}=4$) y otros similares en tamaño es

mayor de la disponible en ambas FPGAs. Esto hace que el modelo con $n_0=8$, $n_{enc}=2$ sea el óptimo para la XC7Z010, mientras que para el caso de la XC7Z020 es el que tiene $n_0=6$, $n_{enc}=4$.

Conclusiones

Este trabajo representa un primer paso en la implementación de modelos de *deep learning* para el diagnóstico de CVDs, mostrando como la estrategia de reducción del modelo permite su implementación en dispositivos de bajas prestaciones. Los autores agradecen al I3A (Instituto de Investigación en Ingeniería de Aragón) por el uso de su clúster HPC (High-Performance Computing) HERMES.

REFERENCIAS

- [1]. SCHMIDT, S.E., HOLST-HANSEN, C., GRAFF, C., TOFT, E. y STRUIJK, J.J., 2010. Segmentation of heart sound recordings by a duration-dependent hidden Markov model. *Physiological Measurement*, vol. 31, no. 4, pp. 513-529. DOI 10.1088/0967-3334/31/4/004.
- [2]. SPRINGER, D., TARASSENKO, L. y CLIFFORD, G., 2015. Logistic Regression-HSMM-based Heart Sound Segmentation. *IEEE Transactions on Biomedical Engineering*, pp. 1-1. DOI 10.1109/TBME.2015.2475278.
- [3]. OLIVEIRA, J., RENNA, F., MANTADELIS, T. y COIMBRA, M., 2019. Adaptive Sojourn Time HSMM for Heart Sound Segmentation. *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 2, pp. 642-649. DOI 10.1109/JBHI.2018.2841197.
- [4]. SPRINGER, D., TARASSENKO, L. y CLIFFORD, G., 2015. Logistic Regression-HSMM-based Heart Sound Segmentation. *IEEE Transactions on Biomedical Engineering*, pp. 1-1. DOI 10.1109/TBME.2015.2475278

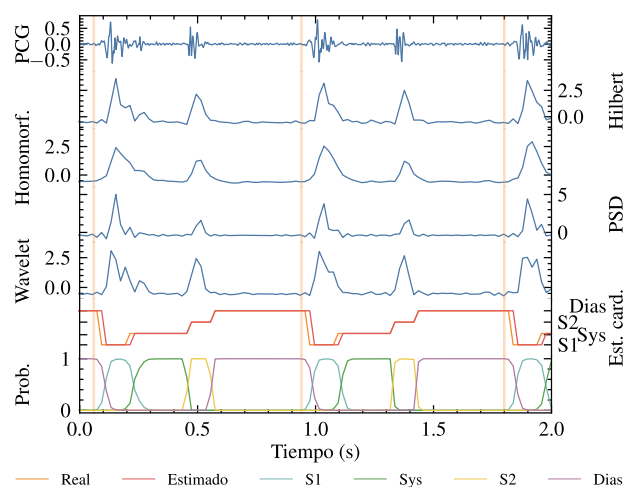


Figura 1. Ejemplo de fonocardiograma y su extracción de los 4 envoltorios. Además, se presentan el estadio cardiaco real y el estimado, junto con las probabilidades que devuelve el modelo para cada instante temporal. Finalmente, en naranja, aparecen las transiciones entre ciclos cardiacos.