MASTER META
EN ELECTRÓNICA Y TELECOMUNICACIÓN APLICADAS
ULPGC
Instituto Universitario de Microelectrónica Aplicada
IUMA

Análisis de las prestaciones de una capa convolutiva de una red neuronal sintetizada en una FPGA mediante herramientas de alto nivel

Adrián Rodríguez Molina armolina@iuma.ulpgc.es
Sebastián López Suárez seblopez@iuma.ulpgc.es
Yubal Barrios Alfaro ybarrios@iuma.ulpgc.es
Román Neris Tomé rneris@iuma.ulpgc.es
septiembre 2021

Resumen:

· A la hora de implementar algoritmos de procesamiento a bordo en los satélites, es común utilizar como plataforma una FPGA, debido a las prestaciones que ofrecen y a que algunas de ellas están certificadas como resistentes a la radiación. En este Trabajo de Fin de Máster se ha hecho uso de una herramienta de síntesis de alto nivel, Vitis HLS, para comprobar cómo afectaba la variación de los parámetros de entrada de una de las capas convolucionales de la red neuronal convolucional Mobilenet a los recursos y latencia de una FPGA. Se partió del código de la capa escrito en Python y se realizó la transcripción de dicho código para generar dos estructuras de capas en C++. A través de Vitis HLS, se han obtenido los resultados de sintetizar varias capas entre las que se variaba el tamaño de la imágen de entrada (aplicándole padding o no), las dimensiones de los pesos y el desplazamiento de estos. A partir de los resultados de la síntesis, se concluye que un tamaño menor de la imagen de entrada produce unos resultados significativamente mejores en términos de latencia y el usar punto fijo reduce de manera más significativa el consumo de recursos.

Número solución	Longitud Imagen	Tamaño Pesos	*Strides*	*Padding*
1	128	3·3	2·2	No
2	128	3·3	2·2	Sí
3	256	3·3	2·2	No
4	256	3·3	2·2	Sí
5	512	3·3	2·2	No
6	512	3·3	2·2	Sí
7	512	4·4	2·2	No
8	512	7·7	1·1	No
9	512	7·7	2·2	No
10	512	7·7	3·3	No
11	512	7·7	4·4	No

Tabla 1: Parámetros de entrada probados en las capas.

Conclusiones:

· El máximo tiempo de ejecución, 1 454,348 ms, se obtuvo al sintetizar la solución 8 de la **Tabla 1** y el mínimo, 3,877 ms, con la solución 1, ambos con la capa *line*.
· El mínimo consumo de memorias, 13 BRAMs se consiguió con la solución 1 de la **Tabla 1**, usando la capa *block*.
· El consumo mínimo de FF y LUTs se obtuvo con la capa *block*, al realizar la síntesis de la solución 5 (899 FFs) y de la solución 6 (2270 LUTs) (**Figura 1**).
· El parámetro que más afecta a la latencia es el tamaño de la imagen y el que más reduce los recursos es el cambio de tipo de dato de punto flotante a punto fijo.

Procedimiento:

· Se realizó la transcripción del código de la capa convolucional de Python a dos estructuras de capas similares en C++, denominadas *block* y *line*, compatibles con la herramienta de síntesis de Vitis HLS.

· Haciendo uso de un script de Python se generaron los ficheros necesarios para realizar la síntesis de las capas de manera que contaran con los parámetros de entrada que se muestra en la **Tabla 1**.

· Se obtuvieron los resultados de latencia y consumo de recursos de la síntesis, comprobando que los resultados de la simulación en C fueran los correctos.

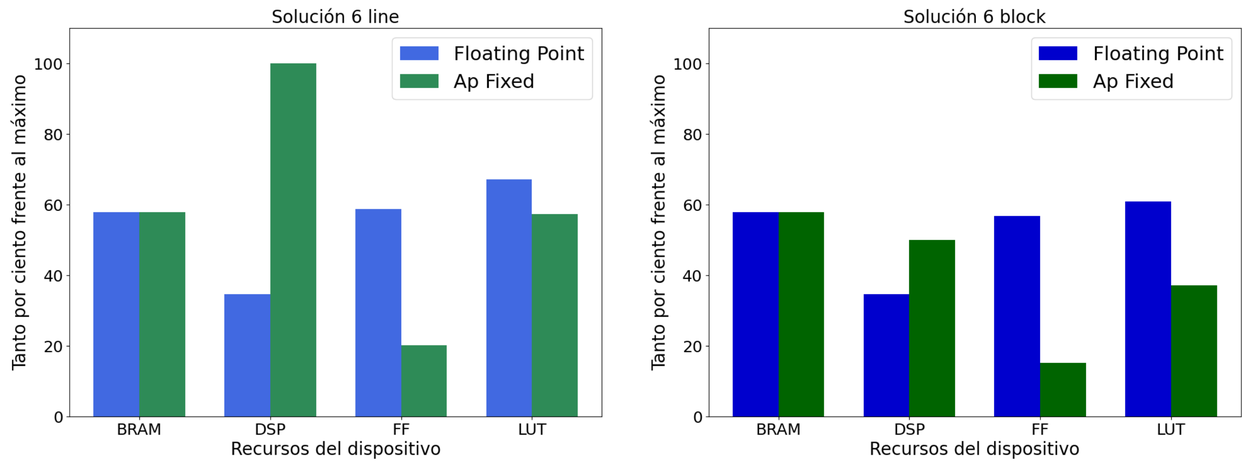

Figura 1: Consumo de recursos de las capas al sintetizar la solución 6 usando punto fijo (En % frente al máximo consumo de recursos obtenidos).