Instituto Tecnológico de Costa Rica

Área Académica de Ingeniería en Computadores

(Computer Engineering Academic Area)

Programa de Licenciatura en Ingeniería en Computadores

(Licentiate Degree Program in Computer Engineering)

Generación de kernels aproximados para algoritmos de aprendizaje de máquina

Informe del Anteproyecto para el Trabajo Final de Graduación

(Report of Pre-project for a Graduation Work in fulfillment of the requirements for the degree of Licentiate in Computer Engineering)

Esteban Calvo Vargas

Cartago, Mayo, 2019

(Cartago, May, 2019)

Tabla de contenidos

1. Palabras clave

Computación aproximada, hardware aproximado, FPGA, aprendizaje de máquina, OpenCL, High-Level Synthesis.

2. Introducción

Las computadoras fueron inicialmente creadas para acelerar los procesos realizados manualmente. En un principio, las computadoras ofrecían una velocidad muy superior a lo que un ser humano podía lograr en actividades específicas, pero su uso se limitaba a procesamiento de datos relativamente bajo. Conforme se han logrado avances en tecnologías de semiconductores, ha surgido la necesidad de utilizar las computadoras para aplicaciones con un alto nivel de procesamiento, así como un creciente número de datos. Esto ha generado una carrera por mantener un nivel de rendimiento aceptable mientras que el consumo de energía, tiempo y capacidad de procesamiento no aumenten (o incluso sean reducidos). La solución a este creciente problema ha sido por varios años aumentar la capacidad de procesamiento (por ejemplo, aumentando la cantidad de transistores por unidad de área). Sin embargo, esta solución trae consigo muchas consideraciones y problemas, en especial al aumento del consumo energético.

Así, ha surgido un nuevo enfoque para solucionar este problema, la computación aproximada. Esta técnica nace a partir del supuesto de que no todos los resultados que se quieren obtener son exactos y precisos. Muchos datos provienen de fuentes no exactas (sensores, mediciones) o no requieren de un procesamiento preciso (aprendizaje de máquina, recomendaciones al usuario, estadística). Esto es, son tolerantes a errores. La computación aproximada pretende explotar de este tipo de datos para generar algoritmos, lenguajes, compiladores, circuitos y arquitecturas que tienen como objetivo reducir el consumo energético o aumentar el rendimiento al costo de tener un resultado aproximado.

Una de las áreas de investigación más importantes en la actualidad es el aprendizaje de máquina. Su característica principal es la toma de decisiones a partir del procesamiento de una gran cantidad de datos. Estos datos pueden ser información escrita, visual (imágenes, video), auditiva y la retroalimentación de los mismos datos para mejorar el proceso de aprendizaje. Por esta razón, la utilización de computación aproximada para reducir el esfuerzo computacional requerido puede ayudar a avanzar de manera significativa esta área de investigación.

El presente ~~trabajo representa el anteproyecto del proyecto final de graduación de Esteban Calvo Vargas, el cual~~ busca explorar métodos y técnicas para la definición de hardware aproximado en FPGA (field-programmable gate array) de manera que pueda ser utilizado para aplicaciones de aprendizaje de máquina, específicamente en la generación de kernels de hardware por medio de OpenCL. Se espera que este trabajo brinde herramientas útiles para cualquier persona que requiera acelerar energéticamente sus algoritmos de aprendizaje de máquina por medio del uso de FPGAs.

Se inicia con una breve descripción sobre computación aproximada, así como el contexto en el que se desarrollará el proyecto, se hablará de la institución en la que se realizará y los beneficiarios de este. Además, se describirá la metodología a seguir, la solución propuesta inicialmente y las herramientas a utilizar para desarrollar el proyecto.

3. Contexto y antecedentes

3.1 Institución donde se realizará el proyecto

El proyecto se realizará en el Karlsruher Institut für Technologie (KIT), universidad enfocada en el desarrollo de tecnología y ciencia. El KIT surge en 2009 a partir de la unión de la Universidad de Karlsruhe, fundada en 1825 como Universidad Fridericiana, y el Centro de Investigación de Karlsruhe. Se ubica en Karlsruhe, en el estado de Baden-Württemberg, al suroeste de Alemania. En el mapa de la figura 1 se puede observar la ubicación del estado Baden-Württemberg dentro del país europeo, así como la ubicación de Karlsruhe dentro de dicho estado.

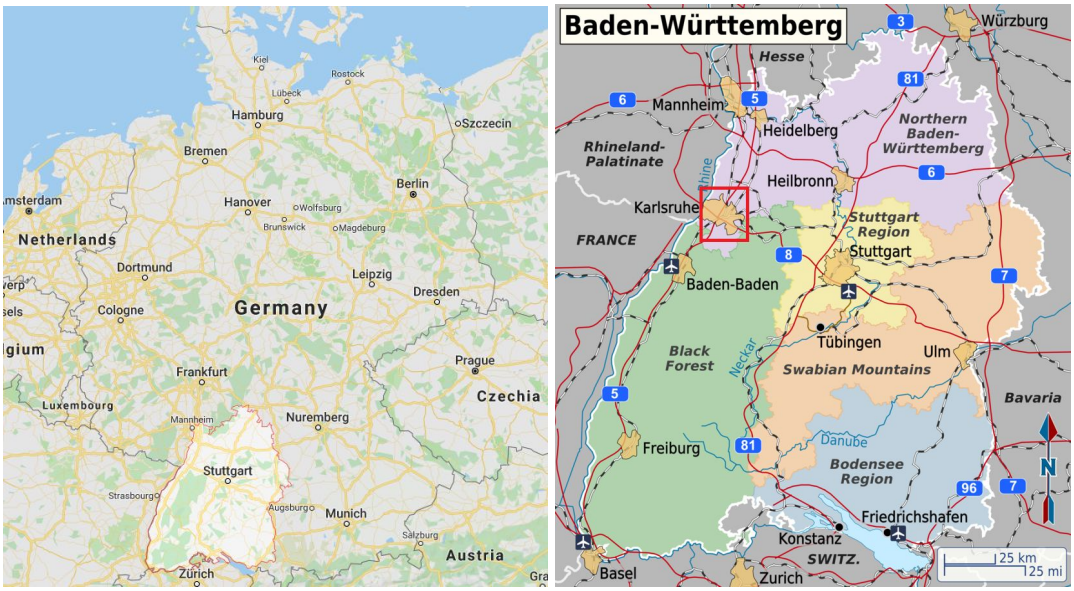

Figura 1. Mapa de Alemania con Baden-Württemberg marcado (izquierda) y localización de Karlsruhe dentro del estado (derecha)

Actualmente el KIT es una de las más prestigiosas universidades técnicas de Alemania, especializada en ciencia e ingeniería. En la figura 2 se presenta el logo institucional de la universidad. Dentro de sus facultades se encuentra la Facultad de Informática, una de las primeras en establecerse en Alemania, la cual contempla diversos institutos enfocados en la enseñanza y la investigación de diferentes temas asociados con Informática.

El Instituto de Ingeniería en Computadores abarca grupos de trabajo que abarcan los diferentes niveles de abstracción de sistemas computacionales. En el Chair for Embedded Systems (CES) se investigan diversos aspectos relacionados con el diseño de sistemas embebidos, desde la confiabilidad de circuitos hasta el manejo de potencia en sistemas de múltiples y muchos núcleos. De forma más general, el Instituto Tecnológico de Karlsruhe posee una organización científica dada por disciplinas agrupadas en cinco divisiones:

- División I: Biología, Química e Ingeniería de Procesos.
- División II: Informática, Economía y Sociedad.
- División III: Ingeniería Mecánica y Eléctrica.
- División IV: Entorno Natural y Construido.
- División V: Física y Matemática.

Figura 2. Logo del Karlsruher Institut für Technologie [1]

Estas divisiones están constituidas por institutos del KIT destinados a trabajos de investigación, innovación y docencia. Los departamentos del instituto son responsables de la educación universitaria. En los Centros KIT, se trabaja en temas de investigación e innovación que se superponen por división, apoyando la cooperación interdisciplinaria.

Las Unidades de Servicio del KIT prestan servicios eficientes para apoyar a los actores en investigación, educación e innovación en el cumplimiento de sus tareas clave. Las Unidades de Servicio del KIT apoyan al personal y a la organización y conciben estrategias de desarrollo en ambas áreas. El detalle de la asignación de tareas dentro de la organización científica se puede observar en la figura 3.

El proyecto como tal será desarrollado en el Chair for Embedded Systems (CES) bajo la tutela de M.Sc. Jorge Alberto Castro Godínez, ingeniero en electrónica, investigador y estudiante de doctorado, quien es egresado del Tecnológico de Costa Rica y posee más de tres años y medio como investigador en el KIT.

3.2 Área de conocimiento

El proyecto se va a realizar dentro del área de computación aproximada, la cual es parte de las áreas de interés de la ingeniería en computadores. Computación aproximada busca relajar la equivalencia entre la especificación y la implementación de aplicaciones tolerantes a errores, esto lo hace por medio de la introducción de errores dentro de un margen aceptable mejorando así la eficiencia energética y un mejor rendimiento. [2]. Dentro de computación aproximada, el proyecto se enfoca en la utilización del FPGA SDK para OpenCL de Intel para el desarrollo de kernels de hardware aproximados y su aplicación en algoritmos de aprendizaje de máquina, evidenciando el efecto que estos kernels tengan en la mejora

de rendimiento y uso energético. Ya que se buscará la definición de hardware aproximado, el trabajo se realizará a nivel de software.

El proyecto se divide en tres partes ordenadas de forma secuencial:

1. Describir el proceso de generación de hardware a partir de la herramienta OpenCL en FPGA de tal manera que se pueda encontrar formas de generar kernels aproximados en vez de exactos.
2. Generar un conjunto de kernels aproximados que puedan ser utilizados en aplicaciones ya existentes de aprendizaje de máquina, caracterizándolos en términos de área, potencia y eficiencia.
3. Evaluar la mejora energética o de rendimiento al aplicar algoritmos de aprendizaje utilizando kernels aproximados en vez de kernels exactos. Esta mejora implica que el resultado está dentro de un rango de error tolerado.

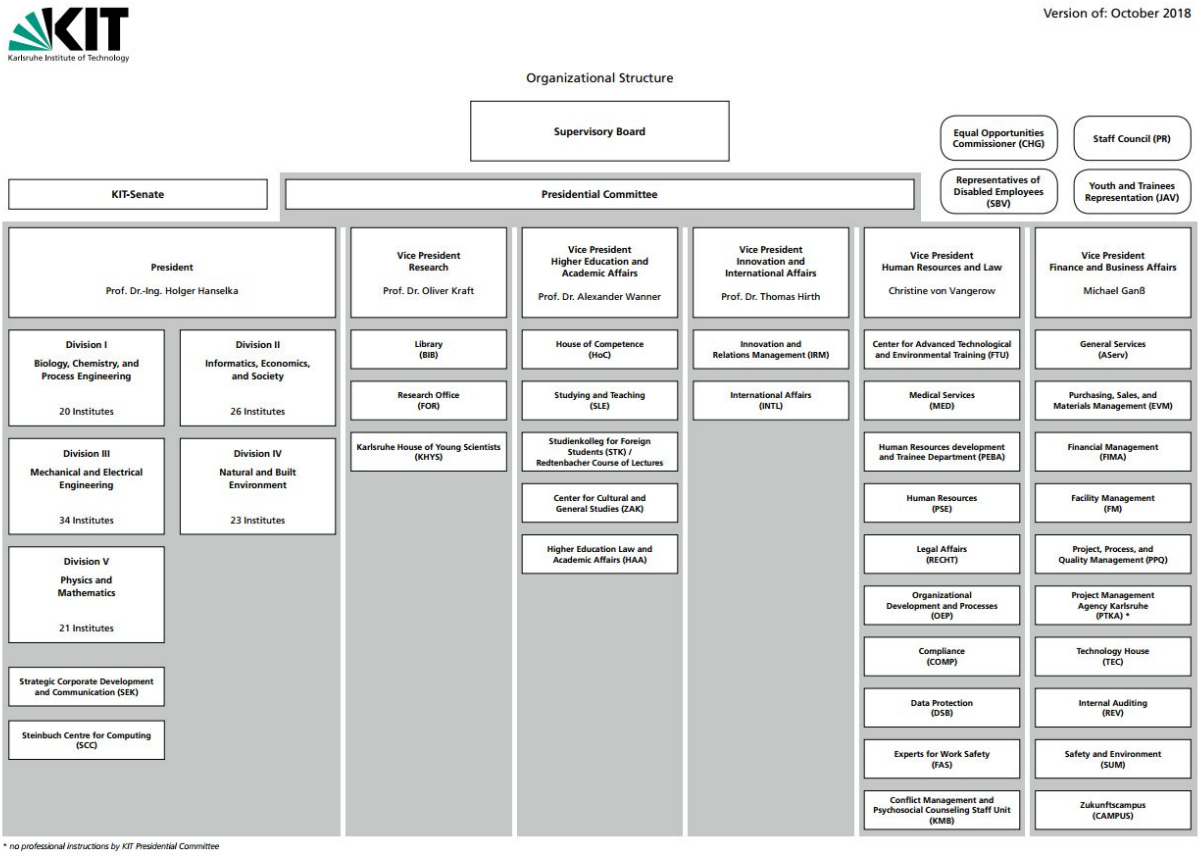

Figura 3. Estructura organizacional del KIT [3]

4. Descripción de la propuesta

4.1 Justificación y definición del problema

4.1.1 Contexto del problema

En los últimos años, han surgido discusiones sobre los límites físicos (y económicos) de la integración a gran escala de transistores [4][5], donde enunciados como la Ley de Moore ejercen presión a las grandes empresas que manufacturan chips. El mismo Gordon Moore ha asegurado que esta tendencia no se puede mantener por mucho tiempo [6]. Esto lleva a la búsqueda de nuevos paradigmas o técnicas que permitan sostener los altos requerimientos de rendimiento y el consumo creciente de energía de las aplicaciones del mundo tecnológico, donde se observa una tendencia al procesamiento cada vez mayor de grandes cantidades de datos. Han empezado a surgir soluciones a esta demanda creciente como lo son los computadores multinúcleo, las arquitecturas multihilo, las computadoras de gran escala, procesamiento por medio de GPU y otros.

A pesar de estas soluciones, existen otros problemas que no se solucionan simplemente con mejorar la arquitectura del procesador. Entre estos problemas están:

- La pared de memoria: Wulf y McKee[7] describen el problema inminente en el que el crecimiento superior de la velocidad de los procesadores supera el crecimiento de las tecnologías en memoria. Esto requiere soluciones que busquen reducir la cantidad de accesos a memoria.
- La pared de utilización: Taylor et al.[8] notaron un fenómeno que se da conforme aumenta la cantidad de transistores en un chip. Este supone un problema en el que al aumentar la escala de integración, el porcentaje utilizable del chip se reduce exponencialmente.
- Problemas de disipación térmica: con el aumento de la frecuencia de los procesadores, el nivel de disipación de calor ha aumentado, forzando una reducción en la tensión de operación. Se han propuesto soluciones a este problema como la utilización de procesadores multinúcleo con núcleos que se desactivan para reducir la carga de trabajo [9].

Debido a estos problemas y a la creciente existencia de aplicaciones resistentes a errores (e.g. [10]) es que surge el paradigma de computación aproximada. Este paradigma busca eliminar el requisito de precisión durante el procesamiento con el objetivo de obtener ganancias de eficiencia energética y velocidad de procesamiento.

Una de las áreas de mayor interés en la actualidad es el del aprendizaje de máquina. Esta área busca la realización de tareas por parte de un sistema computacional sin necesidad de tener una programación específica y, en algunos casos, utilizando retroalimentación a partir de conclusiones previas. Los algoritmos utilizados en el aprendizaje de máquina busca construir modelos matemáticos basados en datos de "entrenamiento" para realizar tareas sin programación explícita. [11] Debido a su naturaleza, las aplicaciones de aprendizaje de máquina no presentan una respuesta exacta inmediatamente, sino

que requieren múltiples ciclos de retroalimentación (y un aumento en los datos de entrenamiento, de ser necesario) para llegar a la respuesta esperada, lo cual quiere decir que computación aproximada puede ser un paradigma para trabajar este tipo de aplicaciones. [12]

4.1.2 Especificación del problema

Las aplicaciones de aprendizaje de máquina se basan en distintos métodos para obtener resultados. Una de las técnicas más utilizadas son las redes neuronales (neural networks) cuyo objetivo es imitar el trabajo que realiza el cerebro humano para obtener resultados. Además, existe una rama dentro del aprendizaje de máquina conocida como aprendizaje profundo (deep learning). Esta consiste en el aumento en la cantidad de capas de aprendizaje en las redes neuronales para obtener una mayor cantidad de detalles a partir de una entrada de datos. [13] Las capas están representadas por nodos (neuronas) que realizan procesamiento sobre la entrada. Este proceso requiere de un alto nivel de procesamiento, pero sus resultados son en su mayoría aproximaciones.

Además, las field-programmable gate array (FPGA, por sus siglas en inglés) son dispositivos que recientemente han sido el objeto de investigaciones con el objetivo de acelerar el procesamiento realizado. Esto debido a que los CPU de uso general no ofrecen la capacidad de procesamiento suficiente (operaciones múltiples con alto nivel de complejidad al mismo tiempo) y los GPU, a pesar de tener capacidades de procesamiento sobre múltiples datos, no son por completo especializados o customizables. El uso de un dispositivo que puede ser programado para tareas específicas (en este caso, redes neuronales) ofrece posibilidades de procesamiento que superan aún a los GPU. [14]

De esta manera, el proyecto surge con el objetivo de aportar a la investigación en el área de FPGAs para uso en aplicaciones de aprendizaje de máquina por medio de definición de hardware aproximado. Cada neurona en una red neuronal es representada por medio de un “kernel” de hardware, que define por sí mismo los cálculos y procesos que debe realizar cada una de las neuronas. Por medio de herramientas de software como OpenCL, es posible definir kernels para realizar tareas de aprendizaje de máquina. Así, combinando todas estas áreas y herramientas, se espera lograr mejoras en rendimiento y en consumo energético necesarios para suplir la demanda de procesamiento actual.

4.1.3 Justificación de la necesidad

La computación aproximada es un área que se encuentra en un período de auge. Existen diversas investigaciones y diseños que buscan aprovechar la existencia de aplicaciones tolerantes a errores. Sin embargo, es necesario continuar avanzando el paradigma para poder observar sus aportes en el día a día. La importancia de este paradigma reside en que no depende del estado actual de la tecnología para brindar mejoras energéticas y de rendimiento.

El uso de FPGA en el área de computación aproximada se encuentra poco explorado y aplicaciones en lenguaje de máquina son de interés global. Una mejora significativa en estos dos campos (y su combinación) generaría nuevas posibilidades para la exploración de soluciones con bajo consumo energético y alto nivel de adaptabilidad.

Así, este proyecto es importante por dos razones:

- Permite avanzar la investigación en el área de definición de hardware aproximado utilizando herramientas populares como OpenCL, lo cual puede servir de base para próximas investigaciones con aplicaciones basadas en FPGA. El uso de una herramienta popular permite agilizar el proceso de generación de resultados en un área poco explorada.
- Genera herramientas listas para ser utilizadas en aplicaciones de aprendizaje de máquina basadas en redes neuronales. Estas pueden ser kernels o conjuntos de kernels customizables. Cualquier interesado que quiera realizar procesamiento y tenga un margen de error tolerable debería ser capaz de utilizar los resultados del proyecto.

4.2 Especificación de objetivos

4.2.1 Objetivo general

Definir un procedimiento para la definición de hardware aproximado en FPGA especializado en algoritmos de aprendizaje de máquina por medio del FPGA SDK para OpenCL de Intel.

4.2.2 Objetivos específicos

- Reconocer el proceso por el cual se describe hardware a partir de la herramienta OpenCL y describir los cambios realizables para generar hardware aproximado en FPGA.
- Crear kernels de hardware aproximados y reutilizables para aplicaciones de aprendizaje de máquina utilizando OpenCL.
- Determinar el nivel de tolerancia al error de aplicaciones de aprendizaje de máquina con redes neuronales al utilizar kernels aproximados en vez de kernels exactos.
- Validar la mejora en tiempo de procesamiento de la utilización de kernels aproximados con respecto a la utilización de kernels exactos tradicionales.

4.3 Beneficios y beneficiarios

Esteban Calvo Vargas: estudiante de la carrera de Ingeniería en Computadores. El interés principal es la finalización de los estudios y la obtención de un título de ingeniero en computadores con grado académico de licenciatura. Además de esto, este proyecto es una oportunidad única de realizar el trabajo final de graduación en una universidad prestigiosa fuera del país, lo cual le va a permitir conocer nuevas culturas y obtener una visión más amplia del mundo. Por último, los conocimientos que se van a adquirir durante el desarrollo del proyecto le permitirán encontrar nuevas oportunidades en un futuro.

Jorge Castro-Godínez, M.Sc.: asesor del proyecto a realizarse en el Karlsruhe Institut für Technologie (KIT). Es investigador y estudiante de doctorado en la universidad alemana, donde tiene a cargo parte de los trabajos de investigación del CES. Dichos trabajos requieren mano de obra e interesados en participar de su desarrollo, necesidad que disminuirá con el presente proyecto. Parte de los beneficios están dados por los aspectos positivos que se obtendrán mediante la interacción tutor-estudiante, con el intercambio de ideas y conocimientos que ayuden a fortalecer sus habilidades técnicas en electrónica y computación.

Chain of Embedded Systems (CES): institución donde se desarrollará el proyecto. Es aquí donde se desarrollan proyectos de investigación e innovación que contribuyen al prestigio del KIT. La investigación y la herramienta que se proponen en este proyecto podrá contribuir al proceso experimental del CES, beneficiando al área de tecnologías de la información.

Ingeniería en Computadores, ITCR: carrera de la cual se está graduando el estudiante. Para la carrera es importante contar con estudiantes que realizan su trabajo de graduación en otro país, le trae prestigio tanto a la carrera como a la institución, mejorando la imagen de ambos y generando un mayor interés por la carrera.

4.4 Supuestos y limitaciones

SUP-01: el estudiante iniciará el proyecto en el mes de Julio en Costa Rica con la fase de investigación teórica. Esto significa que el tiempo para completar el proyecto es de 4 meses.

SUP-02: se estima que 3 meses y medio en Alemania es tiempo suficiente para completar el proyecto. A pesar de que el tiempo de proyecto oficialmente inicia el 22 de Julio, el estudiante estará localizado en Alemania desde el 1ero de Agosto hasta el final de semestre.

SUP-03: el profesor guía se encargará de revisar avances del proyecto y estará dispuesto a colaborar de manera remota con las dudas que surjan durante el desarrollo del proyecto.

SUP-04: el estudiante cuenta con todas las herramientas de software y hardware suficientes para completar el proyecto.

SUP-05: existen trabajos previos que desarrollan el tema de la definición de hardware exacto en FPGA especializado en aprendizaje profundo y redes neuronales. Estos trabajos pueden ser utilizados como base para la generación de hardware aproximado en FPGA.

SUP-06: es posible realizar modificaciones a herramientas de definición de hardware que permitan aprovechar la tolerancia a errores y añadir opciones de computación aproximada a las definiciones obtenidas.

SUP-07: no existen conflictos de propiedad intelectual, confidencialidad o acceso a la información con respecto a los recursos que se necesiten o se tienen disponibles para la realización del proyecto.

LIM-01: el estudiante puede movilizarse a Alemania hasta el primero de Agosto debido a trámites de visa europea.

LIM-02: el presupuesto del estudiante es de 1000 euros mensuales durante la estadía en Alemania.

LIM-03: el estudiante está atado a las regulaciones y limitaciones de la universidad KIT con los estudiantes internacionales.

LIM-04: no hay posibilidad de realizar reuniones presenciales con el profesor guía del proyecto debido a la localización del estudiante durante el proyecto. Toda reunión o comunicación se realizará de manera remota.

4.5 Análisis de riesgos

RIE-01: Recibir un rechazo de la solicitud de la visa. El proceso de solicitud de visa ya fue iniciado, pero existen diversos factores que pueden hacer que la visa sea rechazada y que están fuera del control del estudiante.

- Probabilidad: media. Existe un antecedente de la extensión de solicitud de visa por parte de otro estudiante que realizó el mismo viaje hacia Alemania.
- Impacto: medio. El no tener visa significa que el tiempo de estadía en Alemania se reduce a 3 meses. El tiempo no se reduce demasiado, pero eso añade una presión extra al estudiante y reduce el tiempo efectivo de desarrollo del proyecto.
- Acciones mitigadoras: se va a iniciar el proyecto con la sección de investigación un tiempo antes del viaje hacia Alemania para reducir el impacto de una reducción del tiempo de estadía.

RIE-02: No conseguir un lugar de residencia para la fecha de llegada a Alemania. Los procesos de obtención de residencia en Alemania contienen diversos pasos, entre ellos una entrevista personal que supone una mayor dificultad para obtener un lugar de estadía.

- Probabilidad: baja. A pesar de que cada lugar tiene diferentes procesos, existen muchas opciones de estadía y el precio de alquiler no supone un riesgo para el proyecto. Además, existen opciones para alojarse mientras se busca una estadía permanente.
- Impacto: bajo. De no encontrar un hospedaje para la fecha de llegada, el estudiante deberá disponer de tiempo de desarrollo del proyecto para conseguir el hospedaje.
- Acciones mitigadoras: el estudiante debe agotar todas las opciones existentes de hospedaje meses antes del viaje a Alemania para aumentar las posibilidades de conseguir alojamiento.

RIE-03: El estudiante deberá obtener un vuelo que se adapte a las necesidades de fecha de inicio de las tareas en Alemania y a las limitaciones de tiempo impuestas por la visa (o falta de ella). Debido a que los vuelos suponen un complejo sistema de escalas y destinos, existe un riesgo de obtener un vuelo que llegue a Alemania en una fecha posterior a la prevista.

- Probabilidad: baja. Existen múltiples opciones para conseguir vuelos que se adapten a las diferentes necesidades de las personas.
- Impacto: baja. El proyecto se podría retrasar varios días.
- Acciones mitigadoras: se debe obtener un vuelo que satisfaga las necesidades del estudiante con anticipación y estar atento a nuevas opciones.

RIE-04: La universidad KIT permite a todo estudiante admitido ser registrado en ella. Esto ofrece beneficios específicos para estudiantes matriculados en universidades alemanas, como descuentos en transporte público. El riesgo está en que el registro no pueda ser completado.

- Probabilidad: baja. El estudiante ya fue admitido en la universidad y el cumplir con todos los requisitos reduce la posibilidad de que el registro sea rechazado.
- Impacto: baja. El no ser registrado podría provocar un aumento en los gastos financieros del estudiante,. Sin embargo, esto no supone mayor inconveniente.

- Acciones mitigadoras: el estudiante ha ahorrado dinero extra en caso de necesitar realizar gastos mayores de los esperados.

RIE-05: El proyecto depende de que la herramienta de software permita modificaciones adecuadas para la definición de hardware aproximado en FPGA. Un riesgo del proyecto es que la herramienta no permita realizar definición de hardware no exacto.

- Probabilidad: media. La herramienta a utilizar es Intel® FPGA SDK for OpenCL™. Este es un framework de código-cerrado.
- Impacto: medio. De no ser capaz de utilizar OpenCL para desarrollar el proyecto, se deberá acudir a otras opciones, esto retrasaría el proyecto.
- Acciones mitigadoras: el estudiante deberá buscar otras opciones antes de iniciar el proyecto para evitar un bloqueo en el proyecto.

RIE-06: De ser capaz de realizar modificaciones en la definición de hardware obtenida por parte de la herramienta, existe el riesgo de que estas modificaciones no sean suficientes para obtener hardware aproximado para kernels de redes neuronales.

- Probabilidad: media. Esta es una de las mayores incertidumbres del proyecto y parte de los resultados de la investigación teórica.
- Impacto: alto. El no ser capaz de realizar modificaciones para obtener hardware aproximado significa un replanteamiento por completo del proyecto.
- Acciones mitigadoras: la primera tarea que debe realizar el estudiante es investigar las diferentes modificaciones que se deben realizar para reducir la incertidumbre del proyecto.

RIE-07: Con la suposición de que es posible aproximar el hardware generado por la herramienta de software, aparece el riesgo de que la aproximación realizada en redes neuronales no permita tener un error suficientemente aceptable en los resultados prácticos de los algoritmos de aprendizaje.

- Probabilidad: baja. Existen diversos estudios que tratan el tema de aproximación de redes neuronales.
- Impacto: medio. Aunque no se obtenga un resultado positivo en algoritmos de aprendizaje, el proyecto puede tener resultados que permitan potenciar otras investigaciones.
- Acciones mitigadoras: el estudiante deberá mantenerse informado con respecto a las posibilidades de mejora en redes neuronales así como los niveles de error que pueden ser considerados como aceptables.

5. Propuesta metodológica

5.1 Tipificación del trabajo a realizar

El proyecto consta de dos partes secuenciales. La primera es una investigación teórica sobre el estado de arte de las redes neuronales implementadas en FPGA, la utilización de OpenCL para definir hardware en FPGAs de Intel y la computación aproximada en algoritmos de aprendizaje profundo. La segunda parte consiste en la creación de kernels aproximados y la utilización de estos en aplicaciones de aprendizaje profundo utilizando ejemplos prácticos y demostrando que los resultados son aceptables. Además, se debe demostrar la relación entre el error de los resultados y la mejora de rendimiento de la solución implementada.

5.2 Descripción del proceso a realizar

Se estima que empezando el primero de Julio, el estudiante tendrá 18 semanas de trabajo. Tomando en cuenta 5 días por semana y 6 horas efectivas de trabajo por día, se estima un total de 540 horas. Es recomendado dejar un tiempo provisional por cualquier imprevisto, que va a ser de un 10%. Esto deja al estudiante con una cantidad total de 486 horas planeadas. En la tabla 1 se pueden observar las principales tareas a realizar separadas en 2 grandes categorías y con la asignación de horas estimada.

5.3 Herramientas que se utilizarán

Software

- El sistema operativo a utilizar es Ubuntu 18.04 LTS. Esta versión tiene soporte activo, es simple de instalar y utilizar, además de ser popular por lo que es fácil conseguir soluciones a los problemas que puedan aparecer.
- La herramienta de software para definición de hardware es Intel FPGA SDK for OpenCL™ 18.1. Esta es la versión de OpenCL modificada específicamente para ser utilizada en FPGAs de Intel.
- La documentación será generada utilizando TeXstudio 2.12.14. Este editor de LaTeX es simple de utilizar y tiene soporte para diferentes distribuciones de Linux.
- El editor de texto a utilizar será Visual Studio Code. Este es un editor de texto con una gran cantidad de herramientas para facilitar el desarrollo, incluyendo compiladores y depuración de código. No se especifica la versión debido a que se actualiza una vez al mes.
- Se utilizará ModelSim para simular y probar los módulos de hardware generados. Este es instalado junto con Quartus, el cual es instalado como parte del Intel FPGA SDK for OpenCL™.

Hardware

- Como interfaz de hardware para realizar la síntesis de la FPGA se utilizará la plataforma de desarrollo DE1-SoC de Terasic. Esta es suministrada por el supervisor.
- La plataforma de desarrollo contiene la FPGA Cyclone V.

Código	Descripción	Horas
TA-01	Investigación sobre definición de hardware en FPGA por medio de OpenCL	30
TA-02	Investigación sobre aprendizaje profundo y redes neuronales.	30
TA-03	Investigación sobre redes neuronales en FPGA.	24
TA-04	Investigación sobre trabajos previos por estudiantes alemanes.	30
TA-05	Investigación sobre modificaciones realizables a nivel de OpenCL.	24
TA-06	Investigación sobre aproximaciones realizables en algoritmos de aprendizaje profundo.	36
TA-07	Preparar la estación de trabajo	12
TA-08	Levantamiento de requerimientos	6
TA-09	Pruebas de definición de hardware con OpenCL	24
TA-10	Pruebas de modificación de la definición de hardware	30
TA-11	Definición de la metodología a utilizar para generar redes neuronales aproximadas.	12
TA-12	Definición de un modelo de redes neuronales aproximadas en FPGA con cálculos de error.	18
TA-13	Aplicación del modelo para generar kernels aproximados en FPGA.	30
TA-14	Diseño de una aplicación de aprendizaje profundo.	18
TA-15	Pruebas de la aplicación con hardware exacto.	12
TA-16	Pruebas de la aplicación con hardware aproximado.	12
TA-17	Documentación de los resultados para comparar las diferencias medibles entre hardware exacto y aproximado.	18
TA-18	Diseño de un conjunto de kernels aproximados reutilizables.	30
TA-19	Reuniones de validación.	30
TA-20	Reuniones de avance.	30
TA-21	Redacción del reporte final de proyecto.	30

Tabla 1. Tareas a realizar por el estudiante

5.4 Descripción de entregables

En la figura 4 se pueden se observa un diagrama con los entregables del proyecto.

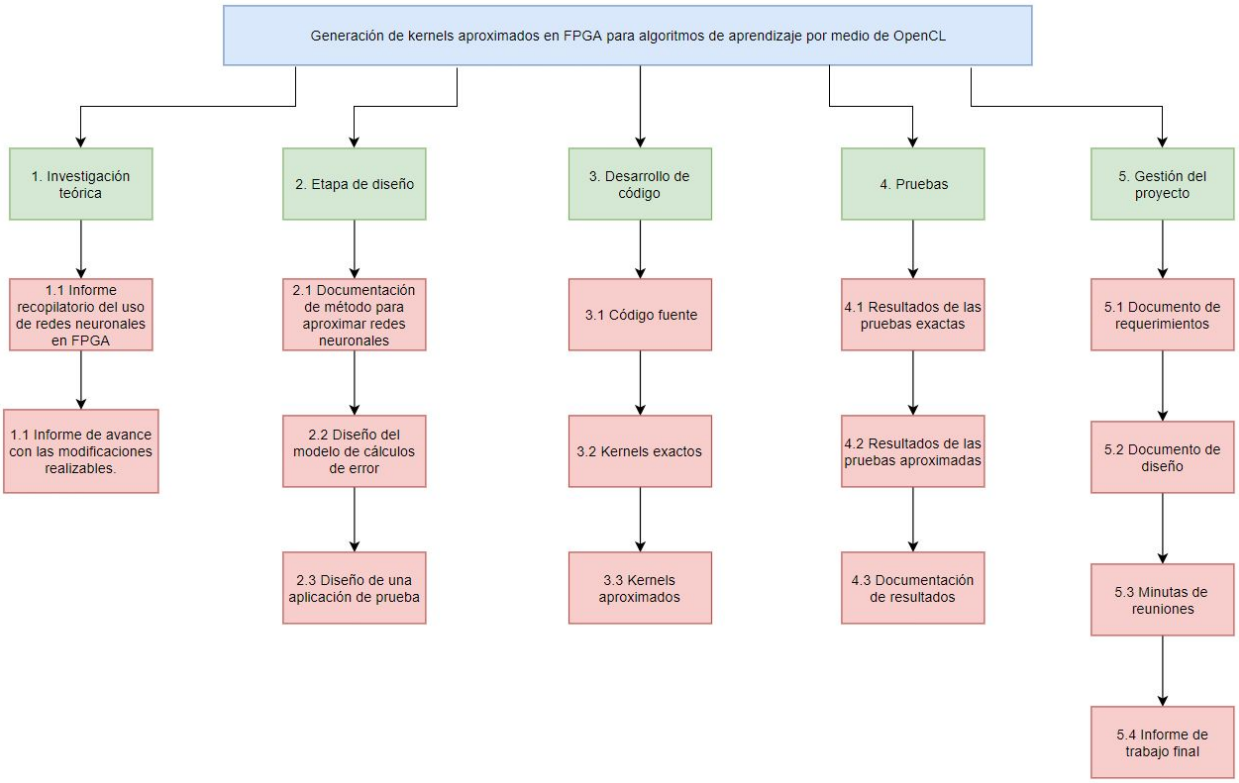

Figura 4. Entregables para el proyecto de graduación

1. Investigación teórica
    1.1. Informe recopilatorio de uso de redes neuronales en FPGA: este informe contendrá toda la información relevante para ser utilizada en el resto del proyecto y que permita tomar decisiones en cuanto a redes neuronales en FPGA.
    1.2. Informe de avance con las modificaciones realizables: informe con las posibilidades de modificación en la herramienta OpenCL y que afecten la definición de hardware en FPGA.
2. Etapa de diseño
    2.1. Documentación de método para aproximar redes neuronales: recopila la información necesaria sobre los principios de computación aproximada que son aplicables en redes neuronales, específicamente para algoritmos de aprendizaje profundo. Debe contener modelos matemáticos que puedan ser comparados contra los resultados del proyecto.
    2.2. Diseño del modelo de cálculos de error: contiene la formulación matemática que permite realizar un cálculo de la precisión de los resultados que se van a obtener y los compare con la ganancia de rendimiento.

2.3. Diseño de una aplicación de prueba: es un documento con el diseño general de una aplicación que será utilizara para realizar las pruebas prácticas una vez se haya desarrollado el proyecto.

3. Desarrollo de código
   3.1. Código fuente: código fuente de la aplicación en OpenCL y cualquier otro código necesario para realizar las pruebas prácticas.
   3.2. Kernels exactos: código fuente de los kernels exactos que se utilizarán para realizar las pruebas.
   3.3. Kernels aproximados: código fuente de los kernels aproximados que se utilizarán para realizar las pruebas.
4. Pruebas
   4.1. Resultados de las pruebas exactas: mediciones de rendimiento y precisión al realizar pruebas sobre kernels exactos.
   4.2. Resultados de las pruebas aproximadas: mediciones de rendimiento y precisión al realizar pruebas sobre kernels aproximadas.
   4.3. Documentación de resultados: comparación de los resultados obtenidos entre las pruebas exactas y aproximadas.
5. Gestión del proyecto
   5.1. Documentación de requerimientos: especificación de los requerimientos completos del proyecto.
   5.2. Documento de diseño: diseño de todas las herramientas desarrolladas durante el proyecto.
   5.3. Minutas de reuniones: contiene toda la información que se obtenga de las reuniones de avance y validación.
   5.4. Informe de trabajo final: informe final del trabajo final de graduación con toda la información relevante que se generó en el proyecto.

4.5 Estrategias de verificación y validación

Todos los entregables del proyecto serán revisados por el supervisor, Jorge Castro Godínez. La estrategia será presentar cada entregable una vez sea obtenido. Esto incluye el informe de trabajo final.

Además, se realizarán diversas pruebas de verificación sobre la aplicación de aprendizaje profundo. Debido a que el proyecto depende de los resultados de estas pruebas, se espera realizar diversas pruebas de cada uno de sus componentes, especialmente los kernels generados. Esto generará datos que se utilizarán para comprobar los modelos matemáticos desarrollados e investigados así como para obtener una conclusión relevante del proyecto.

4.6 Cronograma de trabajo propuesto

En la figura 5 se puede encontrar el cronograma de trabajo utilizando un gráfico Gantt.

ID	Task Name	Duration	Start	Finish
TA-01	Investigación sobre definición de hardware en FPGA por medio de OpenCL / Reunión de avance	6 days	Mon 7/1/19	Mon 7/8/19
TA-02	Investigación sobre aprendizaje profundo y redes neuronales.	5 days	Tue 7/9/19	Mon 7/15/19
TA-03	Investigación sobre redes neuronales en FPGA. / Reunión de validación	5 days	Tue 7/16/19	Mon 7/22/19
TA-04	Investigación sobre trabajos previos por estudiantes alemanes.	5 days	Tue 7/23/19	Mon 7/29/19
TA-05	Investigación sobre modificaciones realizables a nivel de OpenCL. / Reunión de avance	5 days	Tue 7/30/19	Mon 8/5/19
TA-06	Investigación sobre aproximaciones realizables en algoritmos de aprendizaje profundo. / Reunión de validación	7 days	Tue 8/6/19	Wed 8/14/19
TA-07	Preparar la estación de trabajo / Reunión de avance	3 days	Thu 8/15/19	Mon 8/19/19
TA-08	Levantamiento de requerimientos	1 day	Tue 8/20/19	Tue 8/20/19
TA-09	Pruebas de definición de hardware con OpenCL	4 days	Wed 8/21/19	Mon 8/26/19
TA-10	Pruebas de modificación de la definición de hardware	5 days	Tue 8/27/19	Mon 9/2/19
TA-11	Definición de la metodología a utilizar para generar redes neuronales aproximadas. / Reunión de validación	3 days	Tue 9/3/19	Thu 9/5/19
TA-12	Definición de un modelo de redes neuronales aproximadas en FPGA con cálculos de error. / Reunión de avance	4 days	Fri 9/6/19	Wed 9/11/19
TA-13	Aplicación del modelo para generar kernels aproximados en FPGA.	5 days	Thu 9/12/19	Wed 9/18/19
TA-14	Diseño de una aplicación de aprendizaje profundo.	3 days	Thu 9/19/19	Mon 9/23/19
TA-15	Pruebas de la aplicación con hardware exacto. / Reunión de avance	3 days	Tue 9/24/19	Thu 9/26/19
TA-16	Pruebas de la aplicación con hardware aproximado. / Reunión de validación	3 days	Fri 9/27/19	Tue 10/1/19
TA-17	Documentación de los resultados para comparar las diferencias medibles entre hardware exacto y aproximado.	3 days	Wed 10/2/19	Fri 10/4/19
TA-18	Diseño de un conjunto de kernels aproximados reutilizables. / Reunión de validación	6 days	Mon 10/7/19	Mon 10/14/19
TA-21	Redacción del reporte final de proyecto.	5 days	Tue 10/15/19	Mon 10/21/19

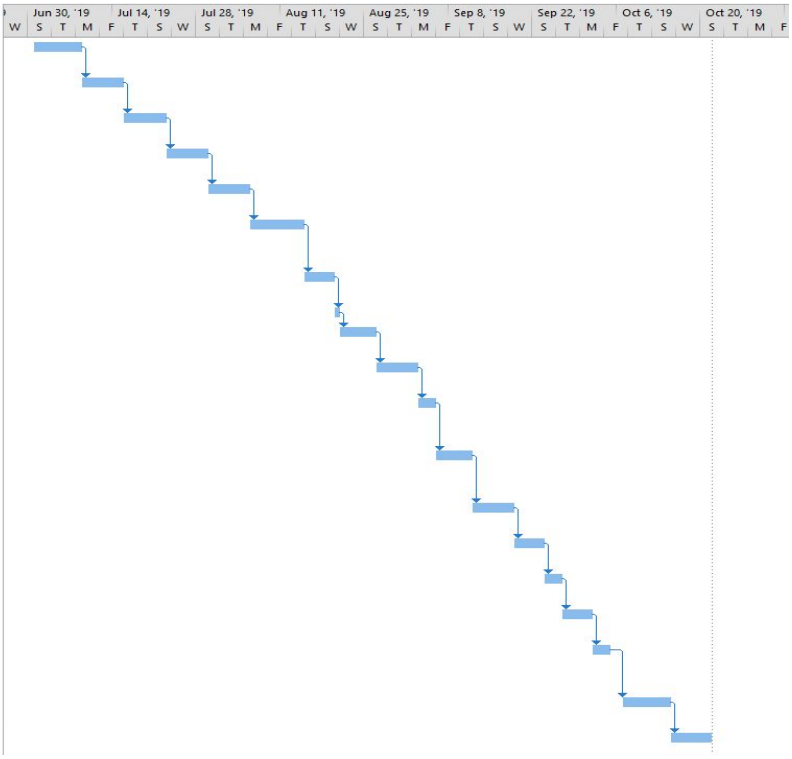

Figura 5. Diagrama de Gantt del trabajo a realizar

Referencias

[1] “KIT logo,” *KIT*. [En línea]. Disponible: http://www.kit.edu/img/intern/10jahre_de.svg. [Accesado: 12-May-2019].

[2] Q. Xu, T. Mytkowicz, and N. S. Kim, “Approximate Computing: A Survey,” *IEEE Design & Test*, vol. 33, no. 1, pp. 8–22, Feb. 2016.

[3] “Organization and Governance,” *KIT*. [En línea]. Disponible: https://www.kit.edu/kit/english/organization.php. [Accesado: 12-May-2019].

[4] “After Moore's law,” *The Economist*, 25-Feb-2016. [En línea]. Disponible: https://www.economist.com/technology-quarterly/2016-03-12/after-moores-law. [Accesado: 12-May-2019].

[5] S. Kumar, Fundamental limits to Moore’s law, arXiv preprint arXiv:1511.05956 (2015).

[6] M. Dubash, “Moore's Law is dead, says Gordon Moore,” Techworld, 10-Mar-2010. [En línea]. Disponible: https://www.techworld.com/news/tech-innovation/moores-law-is-dead-says-gordon-moore-3576581/. [Accesado: 12-May-2019].

[7] W. Wulf and S. McKee, ‘‘Hitting the Memory Wall: Implications of the Obvious,’’ ACM Computer Architecture News 23, No. 1, 20–24 (March 1995).

[8] G. Venkatesh, J. Sampson, N. Goulding, S. Garcia, V. Bryksin, J. Lugo-Martinez, S. Swanson, and M. B. Taylor, “Conservation cores: reducing the energy of mature computations,” *ACM SIGPLAN Notices*, vol. 45, no. 3, 2010.

[9] J. Held et al., “White Paper From a Few Cores to Many: A Tera-scale Computing Research Review,” 2006

[10] D. Thaker, D. Franklin, J. Oliver, S. Biswas, D. Lockhart, T. Metodi, and F. Chong, “Characterization of Error-Tolerant Applications when Protecting Control Data,” *2006 IEEE International Symposium on Workload Characterization*, 2006.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York: Springer, 2006.

[12] A. Agrawal, J. Choi, K. Gopalakrishnan, S. Gupta, R. Nair, J. Oh, D. A. Prener, S. Shukla, V. Srinivasan, and Z. Sura, “Approximate computing: Challenges and opportunities,” *2016 IEEE International Conference on Rebooting Computing (ICRC)*, 2016.

[13] J. Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61: 85–117, 2015

[14] K. Guo, S. Zeng, J. Yu, Y. Wang, and H. Yang, “[DL] A Survey of Fpga Based Neural Inference Accelerator,” *arXiv preprint arXiv:1712.08934*, 2018.

Ingeniería en Computadoras
Ficha de contactos del proyecto

Datos del estudiante

Nombre	Esteban Calvo Vargas
Correo electrónico	estebanedcv@gmail.com
Teléfonos	+506 85569324

Datos del proyecto

Nombre	Generación de kernels aproximados en FPGA para algoritmos de aprendizaje por medio de OpenCL
Breve descripción	El proyecto consta de dos partes: ● Una investigación sobre el uso de FPGAs en aplicaciones de aprendizaje profundo y sobre qué se necesita generar hardware aproximado apropiado para estas aplicaciones. ● El desarrollo de kernels aproximados y la demostración de las pérdidas en precisión pero con la consecuente ganancia en rendimiento para aplicaciones de aprendizaje profundo.
Fecha de inicio	1 de Julio del 2019

Datos de la empresa u organización

Nombre	Chair for Embedded Systems (CES), Karlsruhe Institut für Technologie (KIT), Alemania.
Nombre contacto	Ing. Jorge Alberto Castro Godínez, M. Sc.
Correo electrónico	jorge.castro-godinez@kit.edu
Teléfonos	+49 721 608 46050