

**Programación Concurrente y Distribuida**

**Nombre: Kraenau Rodriguez, Diego Armando (u201710397)**

**Profesor: Carlos Alberto Jara Garcia**

**Trabajo: Trabajo Parcial**

**Sección:** [**CC65-2002-WS7A**](https://aulavirtual.upc.edu.pe/webapps/blackboard/content/listContent.jsp?course_id=_252468_1&content_id=_22175539_1)

2020

**Título del Paper:**

A Deep Learning Based Fault Diagnosis Method with Hyperparameter Optimization by Using Parallel Computing

http://ieeexplore.ieee.org.upc.remotexs.xyz/stamp/stamp.jsp?tp=&arnumber=9142182

**Autores:**

**-**CHAOZHONG GUO

-LIN LI w

-YUANYUAN HU

-JIHONG YAN

**Año de publicación:**

10 de julio del 2020

**Motivación:**

Actualmente con la propuesta de la Industria 4.0 mayormente, muchos países de todo el mundo proponen diferentes soluciones teniendo como objetivo principal explorar y promover la fabricación inteligente. Los equipos de uso tecnológico cada vez se han vuelto más precisos y eficientes, por otro lado, también los diversos datos recopilados constantemente después del funcionamiento de estos equipos, es lo que produce el campo del diagnóstico de fallas en la gran cantidad de datos almacenados. Teniendo esto en cuenta se plantea la siguiente solución para implementar un diagnóstico de fallas teniendo en cuenta el uso del Deep Liaoning y la ayuda de la programación Paralela.

**Problema:**

Actualmente el diagnóstico de fallas tradicional lo pueden realizar principalmente técnicos profesionales y expertos en el diagnóstico, obviamente esto requiere de mucho conocimiento y la experiencia por parte de ellos. Adicionando, que es un poco surrealista que estas personas analicen miles de datos masivos de diversos equipos sin utilizar métodos avanzados. Por lo cual esto genera diferentes fallas que pueden causar problemas costosos a nivel de la industria, desviaciones en la producción e incluso víctimas humanas.

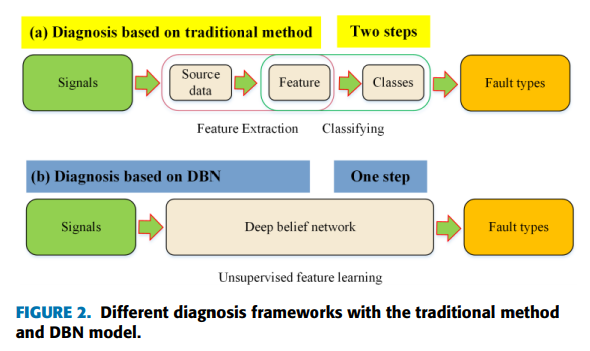
**Propuesta de solución:**

Con lo redactado en la sección anterior, se pondrá en práctica en uso del Deep Learning ya que esto se ha usado ampliamente para el procesamiento del habla y el lenguaje, reconocimiento de imágenes, y también para el diagnóstico de fallas el cual es nuestro problema principal. Básicamente la red del Deep Learning construye un modelo para simular el aprendizaje humano el cual es una herramienta muy utilizada en la era del bigdata. En el caso de aplicar Deep Learning al diagnóstico de fallas, se debe establecer el mapeo de las características de fallas y los tipos de fallas, por lo cual con esto podremos mejorar el diagnóstico de fallas. Adicionalmente, con el rápido y constante desarrollo del bigdata y la gran cantidad de data que se puede generar, la computación rápida y la capacidad de los algoritmos son demasiados importantes, por el cual para resolver este problema del procesamiento y cálculos numéricos se ha hecho uso de la computación paralela. Por lo cual como se sabe el desarrollo asociado con bigdata, el proceso de la obtención de datos o la formación de DBN (Deep Belief Network) toma mucho tiempo, especialmente para datos masivos. Por este motivo se está introduciendo la programación paralela al proceso de entrenamiento del modelo de diagnóstico de fallas basado en DBN con la finalidad que se logre una velocidad de computación más rápida y esto genere una mayor precisión.

**Algoritmos de Deep Learning usados:**

**-DBN (Deep Belief Network):**

Básicamente la red de creencia profunda o mejor conocida como DBN (Deep Belief Network), es un modelo grafico generativo, o alternativamente un tipo de red neuronal profunda el cual está compuesto por diversas capas de variables latentes con conexiones entre ellas. En el momento de cuando se entrena un algoritmo DBN con un conjunto de ejemplos de entrada de forma no supervisada, este algoritmo DBN puede obviamente aprender a reconstruir las posibles entradas por lo cual tiene la poderosa capacidad de aprendizaje sin supervisión. Esto es de suma vitalidad al introducir DBN al diagnóstico de fallas para sistemas. Adicionando que una DBN se compone de múltiples máquinas de Boltzmann restringidas (RBM) no supervisadas y una red de retro propagación supervisada.

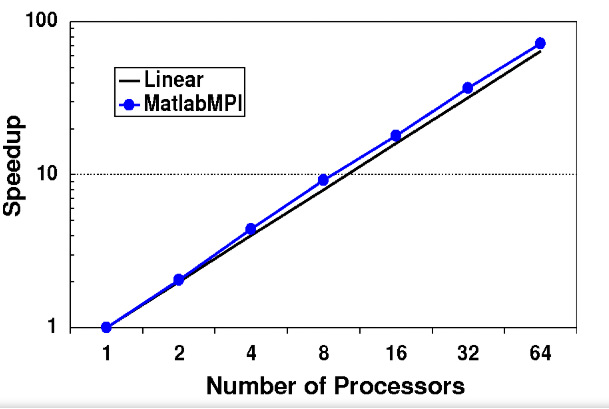


Generalmente, una DBN entrena cada RBM capa por cada, obtenido el hecho de que estas características obtenidas de la capa superior son más representativas que las anteriores. En términos de la última capa, se utiliza básicamente un método supervisado para lograr entrenar la red BP y los errores entre la capa de salida real y la esperada.

**Bibliotecas usadas o descripción de como el algoritmo es paralelizado:**

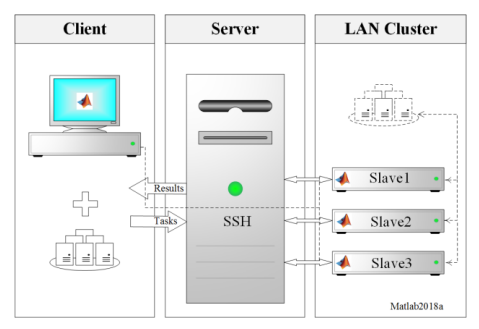
Actualmente ya se posee el algoritmo de Deep Learning, pero el tiempo sigue siendo un problema grave que puede restringir el proceso de entrenamientos de los modelos DBN a medida que se va procesando los datos masivos. Para solucionar este problema se utilizará de computación paralela el cual proporciona un método eficiente para mejorar la velocidad del proceso mediante el uso de múltiples procesadores. La tarea que se va ejecutar se divide en subtareas y cada subtarea se asignara a una unidad de procesamiento independiente. Para este proyecto se utilizará una de las técnicas de la programación paralela llamada MPI, el cual se basa en una interfaz de aplicación de transferencia de información que puede lograr la colaboración de redes de múltiples host para la computación en paralelo.

En la programación de MATLAB, la programación paralela es implementado con la interfaz de paso de mensajes llamada MatlabMPI que se utiliza en el diagnóstico de fallas basado en DBN.



**Arquitectura física o diagrama de infraestructura de la plataforma usada:**

Este paper científico hace un uso completo de los recursos informáticos existentes en su laboratorio, incluidos cuatro servidores multinúcleo de 2U en la misma red de área local (LAN), para realizar el establecimiento de una plataforma de computación paralela distribuida de un clúster de computadoras.



**Opinión critica de su parte (estudiante) de los beneficios o perjuicios del uso de paralelismo en dicho algoritmo:**

El principal beneficio del uso de paralelismo en este proyecto está relacionado a la gran extracción de datos para el posterior análisis de fallas, ya que permite agilizar los procesos de desarrollo por lo que las tareas se dividen en subtareas obteniendo como beneficio la ejecución de instrucciones en simultaneo y la obtención de resultados en menos tiempo.

**Título del paper:**

A deep learning image recognition framework accelerator based parallel computing

http://dl.acm.org.upc.remotexs.xyz/doi/pdf/10.1145/3234804.3234812

**Autores:**

**-** Da Li

- Rui Li

- Shuo Zhang

**Año de publicación:**

20 de junio del 2018

**Motivación:**

Durante la última década, el Deep Learning se ha encontrado en un periodo de aprendizaje y rápido desarrollo, utilizado en diversos campos. En lo general el proceso de capacitación de este modelo se implementará en la nube. En este documento se estableció un marco de reconocimiento de imágenes basado en un acelerador de múltiples núcleos heterogéneo con la finalidad de lograr un generador un proceso de predicción de aprendizaje profundo y mejorar el rendimiento de reconocer imágenes en diversos dispotivios integrados.

**Problema:**

Actualmente, con los avances en redes neuronales de Deep Learning, han abierto demasiadas puertas a una gama de interesantes aplicaciones de procesamiento de imágenes y visión por computadora. El reconocimiento de imágenes se ha aplicado ampliamente en diversos campos, como la navegación y registro fotográfico, etc. El marco de TensorRT proporciona una buena idea para realizar dispositivos integrados diseñados para inferencia, sin embargo, el problema actual que surge gracias al aumento de los requisitos de precisión y la complejidad de las aplicaciones prácticas, el tamaño de las redes de Deep Learning se vuelve cada vez más grande.

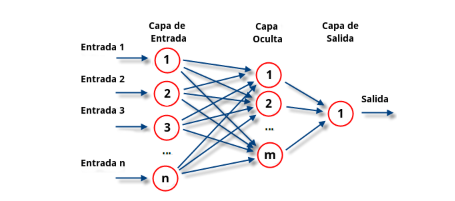
**Propuesta de solución:**

La plataforma de propósito especial es una solución adecuada para que el dispositivo integrado se ocupe de una gran cantidad de cálculos en Deep Learning, pero los campos de aplicación serán obviamente limitados. Por lo tanto, el procesador integrado de múltiples núcleos será una mejor manera de resolver una gran cantidad de cálculos durante el proceso de predicción. El siguiente documento presenta un marco de reconocimiento distribuido escalable para lograr el algoritmo de predicción.

**Algoritmos de Deep Learning usados:**

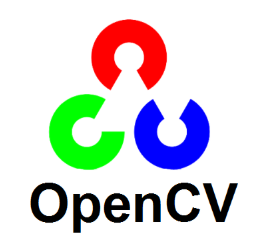
**-RNA (Red Reunoral Artificial):**

Como las redes neuronales tradicionales, el proceso de predicción de la red neuronal calcula capa por capa desde la primera capa de entrada hasta la misma capa de salida y las diferentes capas de salida actual son las entradas de las siguientes capas. Estos sistemas aprenden y se forman a sí mismos, en lugar de ser programador de forma explícita y sobresalen en áreas donde la detección de soluciones o características es difícil de expresar con la programación convencional.



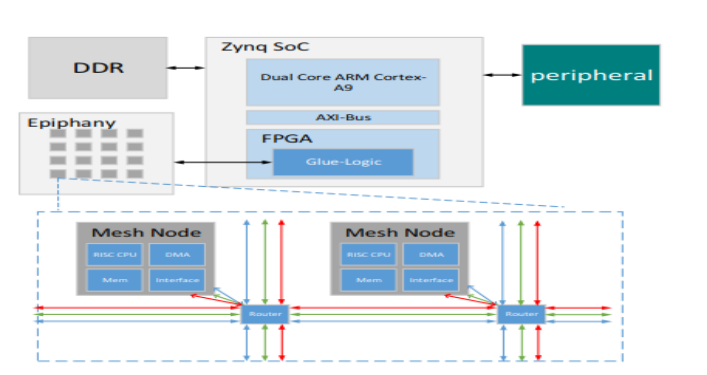
**Bibliotecas usadas o descripción de como el algoritmo es paralelizado:**

Basadas en OpenCV, las bibliotecas subyacentes utilizaron libpng para completar el proceso de pre procesamiento de imágenes, como la eliminación de ruido, etc. Por lo tanto, en este documento también se adoptó libpng para lograr el mismo efecto, almacenar la imagen pequeña original con matrices bidimensionales utilizadas con un numero de 0 a 255.Alcenandolos en conjuntos de data mnist los pesos de las imágenes y los mapas intermedios en memoria en chip. Se adaptó un enfoque de mapas paralelos, donde cada uno de los 16 eCores comparte la carga de trabajo de evaluar todos los mapas en una capa determinada en paralelo.



**Arquitectura física o diagrama de infraestructura de la plataforma usada:**

Epiphany es un procesador multinucleo en chip escalable, que consta de núcleos individuales que tiene una unidad de punto flotante (FPU) y una ALU entera. Todos los núcleos están conectados con una red de malla 2D. Epiphany es una arquitectura de eficiencia energética y eficiente espacial muy adecuada para aplicaciones informáticas integradas de alto rendimiento.



(Arquitectura del sistema Epiphany y Paralela)

**Opinión critica de su parte (estudiante) de los beneficios o perjuicios del uso de paralelismo en dicho algoritmo:**

Uno del principal beneficio del uso de paralelismo en este algoritmo es permitir varias instrucciones en simultaneo y utilizar más que todo el GPU para aumentar el tiempo de respuesta sobre el reconocimiento de imágenes. Ya sea usando bibliotecas como OpenCV para ayudar al pre procesamiento de las imágenes almacenadas.

**Título del Paper:**

TensorLightning: A Traffic-Efficient Distributed Deep Learning on Commodity Spark Clusters

http://ieeexplore.ieee.org.upc.remotexs.xyz/stamp/stamp.jsp?tp=&arnumber=8369060

**Autores:**

-SEIL LEE -CHANG-SUNG JEONG2

-HANJOO KIM -SUNGROH YOON

-JAEHONG PARK

-JAEHEE JANG

**Año de publicacion:**

23 de mayo del 2018

**Motivación:**

Debido al reciente éxito del Deep Learning, big data y los diversos cálculos que siguen creciendo diariamente, genera el amplio tema de investigación de sistemas de auto aprendizaje profundo distribuido. Aunque un entorno distribuido de escalamiento horizontal que utiliza servidores básicos no solo tiene un límite debido a la operación síncrona y el tráfico de comunicaciones, sino que también se plantea en este documento la combinación del entrenamiento de redes neuronales profundas (DNN) con clústeres existentes. Por lo que este enfoque proporciona una rápida formación de DNN y de flexibilidad alta.

**Problema:**

Actualmente como ya se conoce, las Deep Neural Network(DNN) han logrado desempeños notables en diversos ámbitos de problemas de aprendizaje. Básicamente para entrenar eficazmente un modelo grande de DNN, se requiere una gran cantidad de datos. De ahí viene la necesidad de usar un sistema distribuido que gestione tan enormes tareas de cálculo y datos para poder obtener una respuesta certera del algoritmo evaluando los cientos de datos ingresados.

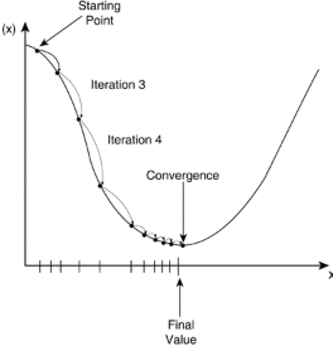
**Propuesta de solución:**

En este documento, proponen el uso de TensoLighthing, un nuevo marco de aprendizaje profundo distribuido para clústeres de productos básicos Spark con el mismo objetivo de superar las limitaciones mencionadas anteriormente. Implementamos el Esqueda de servidor de parámetros para ejecutar el algoritmo. TensorLighthing combina Caffe y TensoFlow, por lo que se puede implementar varios modelos de entrenamiento para el algoritmo, desde modelos básicos hasta modelos de redes neuronales recurrentes. En el siguiente marco de trabajo se evaluó atreves de experimentos en dos modelos representativos de DNN, los cuales son la clasificación de imágenes y modelado secuencia a secuencia.

**Algoritmos de Deep Learning usados:**

**-Stochastic Gradient Descent (SGD):**

Este algoritmo en el Deep learning también es conocido como descenso de gradiente incremental, es básicamente un método iterativo que tiene una finalidad de optimizar una función objetivo diferenciable el cual es una aproximación estocástica de la optimización del descenso de la gradiente. En este documento se modifica un poco el algoritmo para volverlo paralelizado y tener el objetivo de optimizarlo para acelerar los cálculos con los diversos datos de entrada.



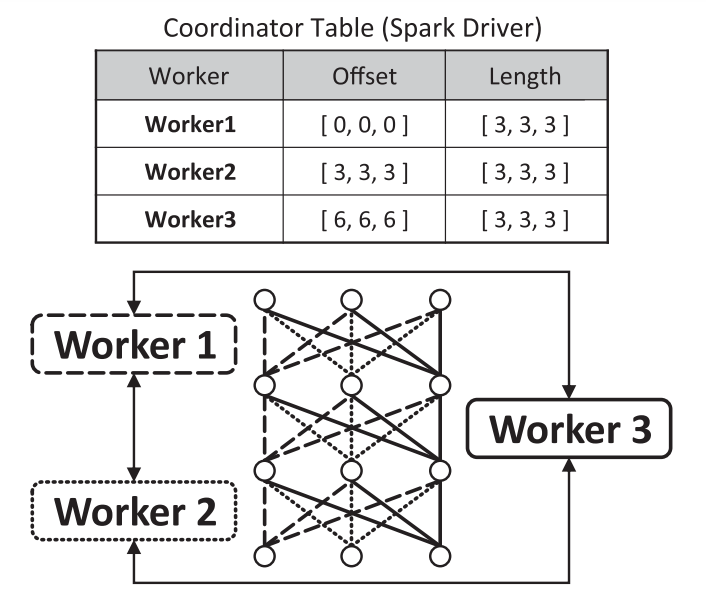
**Bibliotecas usadas o descripción de como el algoritmo es paralelizado:**

Para este documento se definió usar TensorLightning, el cual integra la canalización de datos ampliamente utilizada de Apache Spark con potentes bibliotecas de aprendizaje profundo Caffe y TensorFlow el cual ayudara con el aprendizaje automatico a través de un rango de tareas.



**Arquitectura física o diagrama de infraestructura de la plataforma usada:**

Se creó un entorno de clúster de productos básicos para validar el rendimiento de el algoritmo.



Ejemplo del intercambiador de parámetros distribuidos. Cada nodo trabajador asume la responsabilidad de una parte de los parámetros globales ŵ, respectivamente. Durante la formación, cada nodo trabajador se comunica con los demás según la tabla de coordinadores

**Opinión critica de su parte (estudiante) de los beneficios o perjuicios del uso de paralelismo en dicho algoritmo:**

Uno de los beneficios del paralelismo en este algoritmo es acelerar el procesamiento de la DNN pero dicha desventaja se verá reflejada en el consumo de energía y una breve complejidad construir una buena sincronización. También se evidencia el uso de librerías como TensorFlow y Cafee2 para el aprendizaje autónomo del algoritmo.