Análisis de métodos de inteligencia artificial para la reducción de incertidumbre

Pamela Agustina Chirino, Germán Bianchini, Paola Guadalupe Caymes Scutari

Universidad Tecnológica Nacional-Facultad Regional de Mendoza
pamelaachirino@gmail.com;gbianchini@frm.utn.edu.ar;pcaymesscutari@frm.utn.edu.ar

Abstract
La inteligencia artificial ha generado una revolución importante en el mundo de la computación en los últimos años aplicándose en diversos campos. En este documento se propone estudiar dos formas de inteligencia artificial, como lo son las redes neuronales y la visión computacional, y la posible aplicación del paralelismo en estas mismas para optimizarlas. En este contexto las aplicaremos a un modelo de predicción de incendios, ya existente y llevado a cabo por el Laboratorio de Investigación en Cómputo Paralelo/Distribuido de la UTN-FRM.

Palabras Clave
Inteligencia Artificial, Redes neuronales, Paralelismo, Visión Computacional, Predicción

1. Introducción

La Inteligencia Artificial es la simulación de inteligencia humana por parte de las máquinas. Dicho de otro modo, es la disciplina que trata de crear sistemas capaces de aprender y razonar como un ser humano. Normalmente, un sistema de inteligencia artificial es capaz de analizar datos en grandes cantidades, identificar patrones y tendencias y, por lo tanto, formular predicciones de forma automática, con rapidez y precisión.**[1]**
Con el fin de reducir la incertidumbre en el modelo de predicción de incendios desarrollado en el Laboratorio de Investigación en Cómputo Paralelo/Distribuido se estudiarán a continuación dos métodos de la misma: Redes neuronales y Visión Computacional.
En la Sección 2 y 3 estudiaremos dos técnicas o métodos de la inteligencia artificial, a saber las Redes Neuronales y la Visión Computacional, con la finalidad de analizar las posibilidades que ofrecen para ser incorporadas a una familia de métodos de predicción de incendios forestales **[2]** y así mejorar su rendimiento. En la Sección 4 describiremos la propuesta de paralelización de estas dos técnicas y el trabajo desarrollado en ellas.

2.Redes Neuronales

Las redes neuronales artificiales son un modelo computacional inspirado en el comportamiento observado en el cerebro humano. Consiste en un conjunto de unidades, llamadas neuronas artificiales, conectadas entre sí para transmitir o comunicar señales. La información de entrada atraviesa la red neuronal (donde se somete a diversas operaciones) produciendo valores de salida. Para comprender mejor se comparará, como se muestra en la figura 1, una neurona biológica y una artificial. **[3]**

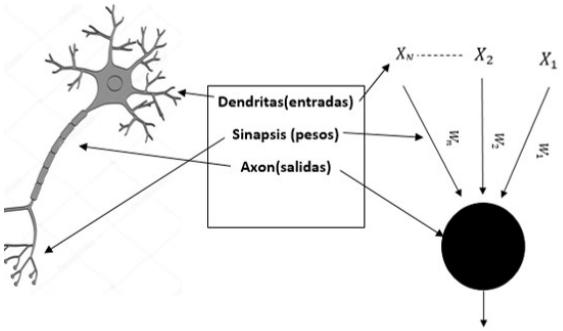

Figura 1:Comparación entre neurona biológica y artificial.

Una neurona biológica está compuesta por el cuerpo de la célula (soma), como se observa en la parte izquierda de la figura y dos tipos de ramificaciones: el axón y las dendritas. La neurona recibe las señales (impulsos) de otras neuronas a través de sus dendritas y transmite señales generadas por el cuerpo de la célula a través del axón.

Para establecer una similitud directa entre la actividad sináptica y las redes neurales artificiales podemos considerar que las señales que llegan a la sinapsis son las entradas a la neurona; éstas son ponderadas a través de un parámetro denominado peso, asociado a la sinapsis correspondiente. Estas señales de entrada pueden excitar a la neurona (sinapsis con peso positivo) o inhibirla (peso negativo), y dichos pesos influirán en la salida de la neurona **[4].**

2.1. Elementos básicos que componen una red neuronal

Una red neuronal artificial está constituida por neuronas interconectadas y arregladas en capas, como se muestra en la Figura 2. Los datos ingresan por medio de la "capa de entrada", pasan a través de la "capa oculta" y salen por la "capa de salida". Cabe mencionar que la capa oculta puede estar constituida por varias capas de neuronas.

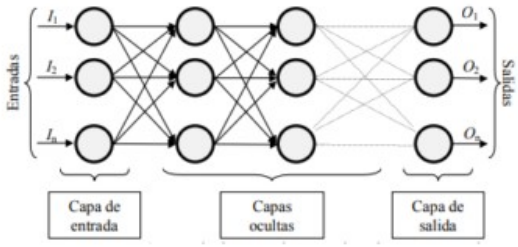

Figura 2: Esquema de una red neuronal.

Además de las capas ya mencionadas, una red neuronal cuenta con las funciones que se detallarán a continuación para poder llevar a cabo su funcionamiento **[4].**

2.2.1. Función de entrada

La neurona trata a muchos valores de entrada como si fueran uno solo; esto recibe el nombre de entrada global. La forma de lograr la entrada global es a través de la función de entrada, la cual se calcula a partir del vector entrada. La función de entrada puede describirse como sigue:

$$input_i = (in_{i1} \bullet w_{i1}) * (in_{i2} \bullet w_{i2}) * \ldots (in_{in} \bullet w_{in})$$

(Ecuación n°1)

donde: * representa al operador apropiado, $\bullet$ es el producto, n al número de entradas a la neurona n_i y w_i al peso.

Los valores de entrada se multiplican por los pesos anteriormente ingresados a la neurona. Estos pesos de los valores de entrada cambian los pesos de las neuronas haciendo que cambie también la influencia de las mismas, es decir, una neurona con mayor peso tendrá mayor relevancia en el problema que una con un peso inferior a esta.

2.2.2. Función de activación

La función de activación tiene como objetivo acotar los valores de salida de una red neuronal para mantenerlos en ciertos rangos, es decir, calcula el estado de actividad de una neurona; transformando la entrada global en un valor (estado) de activación.

2.2.3. Función de salida

El último componente que una neurona necesita es la función de salida. El valor resultante de esta función es la salida de la neurona (out_i); por ende, la función de salida determina qué valor se transfiere a las neuronas vinculadas. Si la función de activación está por debajo de un umbral determinado, ninguna salida se pasa a la neurona subsiguiente.

Lo descripto anteriormente, de los componentes y funciones de una red neuronal, se puede apreciar en la Figura 3.

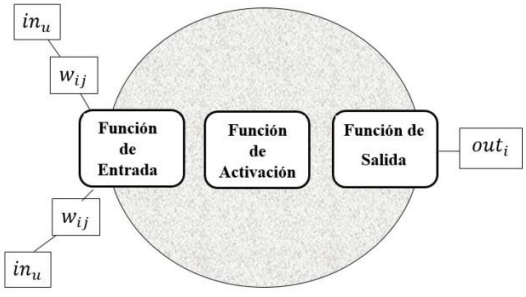

Figura 3: Esquema de una red neuronal y sus funciones.

2.3. Aprendizaje de las redes neurales

El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada. Los cambios

que se producen durante el mismo se reducen a la destrucción, modificación y creación de conexiones entre las neuronas. Una red neuronal debe aprender a calcular la salida correcta para cada arreglo o vector de entrada en el conjunto de ejemplos. En los modelos de redes neuronales artificiales, la creación de una nueva conexión implica que el peso de la misma pasa a tener un valor distinto de cero.
El método de aprendizaje que nos interesa para este artículo es el aprendizaje supervisado, que se caracteriza porque el proceso de aprendizaje se realiza mediante un entrenamiento controlado por un agente externo, que podría ser el programador (supervisor, maestro) que determina la respuesta que debería generar la red a partir de una entrada determinada. El supervisor controla la salida de la red y en caso de que ésta no coincida con la deseada, se procederá a modificar los pesos.
Para el estudio práctico de lo presentado anteriormente, estudiamos y ponemos en práctica este tipo de aprendizaje con la red neuronal Perceptrón.

2.4. Perceptrón

El perceptrón es la red neuronal más básica que existe de aprendizaje supervisado. El funcionamiento del perceptrón es muy sencillo, simplemente lee los valores de entrada, suma todas las entradas de acuerdo a unos pesos y el resultado lo introduce en una función de activación que genera el resultado final. En la Figura 4 se puede observar un esquema del percpetrón.

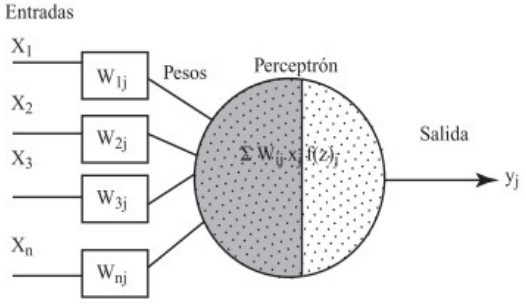

Figura 4:Esquema de Perceptrón.

Un aspecto importante a tener en cuenta es que el perceptrón utiliza un tipo particular de aprendizaje supervisado llamado Retro propagación de error o *backpropagation*.

Es un método de aprendizaje supervisado con descenso del gradiente. En las redes de retropropagación primero se aplica un patrón de entrada, el cual se propaga por las distintas capas que componen la red hasta producir una salida de la misma. Esta salida se compara con la salida deseada y se calcula el error cometido por cada neurona de salida. Estos errores se transmiten hacia atrás, partiendo de la capa de salida hacia todas las neuronas de las capas intermedias. Cada neurona recibe un error que es proporcional a su contribución sobre el error total de la red. Basándose en este error recibido se ajustan los pesos sinápticos de cada neurona **[1].**

Para enseñarle a la red neuronal es necesario entrenar un conjunto de datos, el cual consta de señales de entradas asignadas a una denominada "salida deseada". En el entrenamiento la salida deseada representa la salida esperada para cierto patrón de entrada, es decir, pasamos a la red un conjunto de entradas con la salida que debería producir ese conjunto.
En cada iteración los pesos de los nodos se modifican usando nuevos datos del conjunto para el entrenamiento.
La salida de la red es comparada con la salida deseada. La diferencia entre la salida de la red y la salida deseada se denomina error de la señal. El algoritmo de retropropagación propaga el error de regreso a todas las neuronas, cuya salida fue la entrada de la última neurona. Luego el error se va propagando a las neuronas de capas anteriores, considerando los pesos de las conexiones.
Se considera que la red ha aprendido cuando el error es 0 o un margen próximo al mismo **[5][6]**.

3. Visión computacional

La visión computacional trata de emular en las computadoras la capacidad que tienen nuestros ojos. Es decir, trata de interpretar las imágenes recibidas por dispositivos y

reconocer los objetos, ambiente y posición en el espacio. Debido a que parte de nuestro trabajo se centrará en el tratamiento de imágenes satelitales, hemos considerado estudiar este tema y su posible paralelización para aplicarlo en el modelo de predicción de incendios anteriormente mencionado. Por lo tanto, en la siguiente sección se describirá de manera general el manejo de las imágenes a través de la visión computacional **[7].**

3.1. Niveles de visión computacional

La Visión computacional consiste en partir de una imagen (pixeles) y llegar a una descripción (predicados, geometría, etc.) adecuada de acuerdo a nuestro propósito. Como este proceso es muy complejo, se ha dividido en varias etapas o niveles de visión:

- Procesamiento de nivel bajo: se trabaja directamente con los pixeles para extraer propiedades como orillas, gradiente, profundidad, textura, color, etc.
- Procesamiento de nivel intermedio: consiste generalmente en agrupar los elementos obtenidos en el nivel bajo, para obtener líneas, regiones, generalmente con el propósito de segmentación.
- Procesamiento de alto nivel: está generalmente orientada al proceso de interpretación de lo obtenido en los niveles anteriores y se utilizan modelos y/o conocimiento del dominio.

Estos niveles de procesamiento son llevados a cabo por algoritmos. El algoritmo seleccionado para estudiar en este artículo es el de redes convolucionales **[7]** que se describe a continuación.

Etapas en un proceso de visión artificial

En esta sección se detallarán los pasos fundamentales para llevar a cabo la visión computacional.

El primer paso en el proceso es adquirir la imagen digital. Para ello se necesitan sensores y las capacidades para digitalizar la señal producida por el sensor.

Una vez que la imagen digitalizada ha sido obtenida, el siguiente paso consiste en el procesamiento de dicha imagen. El objetivo del procesamiento es mejorar la imagen de forma que el objetivo final tenga mayores posibilidades de éxito.

El siguiente paso es la segmentación. Su objetivo es dividir la imagen en las partes que la constituyen o los objetos que la forman. En general la segmentación autónoma es uno de los problemas más difíciles en el procesamiento de la imagen.

La salida del proceso de segmentación es una imagen de datos que, o bien contienen la frontera de la región o los puntos de la misma.

La elección de una representación es solo una parte de la transformación de los datos de entrada. Es necesario especificar un método que extraiga los datos de interés. La parametrización, que recibe también el nombre de selección de rasgos, se dedica a extraer rasgos que producen alguna información cuantitativa de interés o rasgos que son básicos para diferenciar una clase de objetos de otra.

En el último lugar se encuentra el reconocimiento y la interpretación. El reconocimiento es el proceso que asigna una etiqueta a un objeto basada en la información que proporcionan los descriptores (clasificación). La interpretación lleva a asignar significado al conjunto de objetos reconocidos.

En la Figura 5 se muestra un diagrama para resumir y entender de manera sintética lo explicado anteriormente.

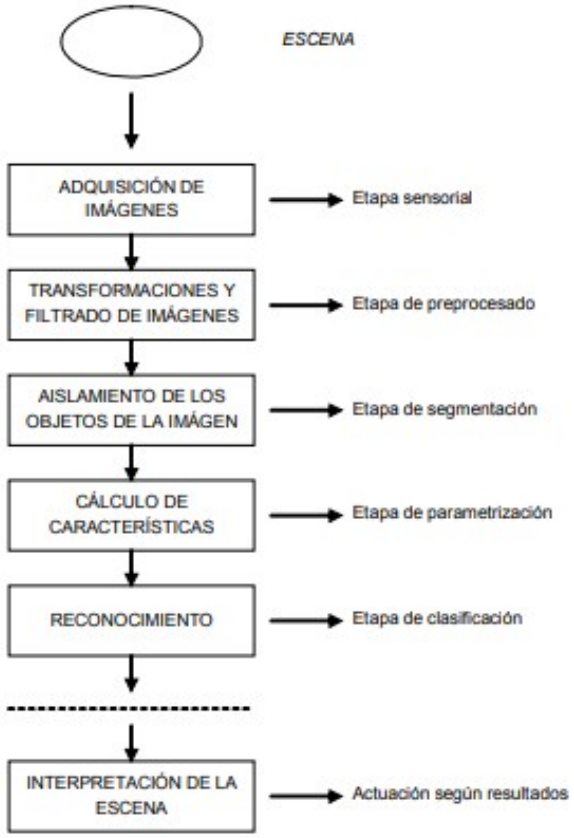

Figura 5: Diagrama

A continuación, se describirá como llevar a cabo este proceso con redes neuronales. Siguiendo la analogía presentada anteriormente, el cerebro humano es el encargado del procesamiento de imágenes y así como existen las redes neuronales para simular su proceso de aprendizaje y decisión, existen las redes convolucionales, presentadas a continuación, para simular su procesamiento de imágenes.

3.2. Redes convolucionales

Las redes neuronales convolucionales son muy similares a las redes neuronales ordinarias como el perceptrón multicapa que fue descripto anteriormente; se componen de neuronas que tienen pesos y capacidad de aprender.

Lo que diferencia a las redes neuronales convolucionales es que suponen explícitamente que las entradas son imágenes, lo que nos permite codificar ciertas propiedades en la arquitectura de las redes neuronales tradicionales, permitiendo ganar en eficiencia y reducir la cantidad de parámetros en la red.

Las redes neuronales convolucionales trabajan modelando de forma consecutiva pequeñas piezas de información, y luego combinando esta información en las capas más profundas de la red.

3.2.1. Estructura de las Redes Neuronales Convolucionales

En general, las redes neuronales convolucionales van a estar construidas con una estructura que contendrá tres tipos distintos de capas que se detallarán a continuación y se pueden observar en la Figura 6.

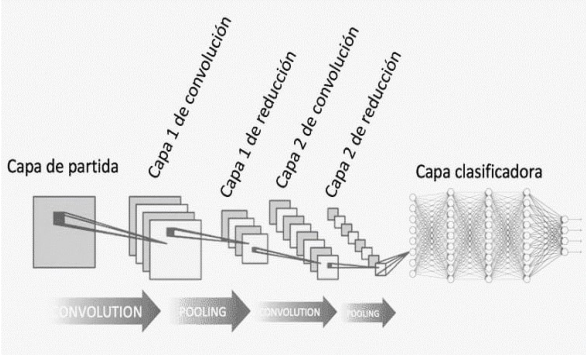

Figura 6: Esquema de la estructura de una red convolucional.

3.2.1.1. Capa convolucional

Esta es la capa que le da nombre a la red, lo que distingue a las redes neuronales convolucionales de cualquier otra red neuronal es que utilizan una operación llamada convolución en alguna de sus capas. La operación de convolución recibe como entrada o *input* la imagen y luego aplica sobre ella un filtro o *kernel* que nos devuelve un mapa de las características de la imagen original, y de esta forma logramos reducir el tamaño de los parámetros.

3.2.1.2. Capa de reducción o *pooling*

La capa de reducción o *pooling* se coloca generalmente después de la capa convolucional. Su utilidad principal radica en la reducción de las dimensiones espaciales (ancho x alto) del volumen de entrada para la siguiente capa convolucional.

En la figura 7 se observa lo explicado anteriormente de manera gráfica.

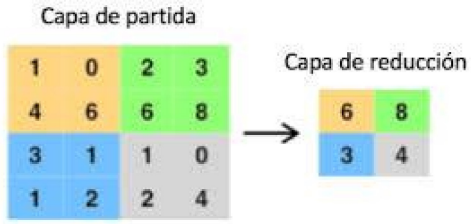

Figura 7: Esquema de la Capa de Reducción.

3.2.1.3. Capa clasificadora totalmente conectada

Al final de las capas convolucional y de *pooling*, las redes utilizan generalmente capas completamente conectadas en la que cada pixel se considera como una neurona separada al igual que en una red neuronal regular.

4. Paralelismo

El procesamiento paralelo es un método mediante el cual se dividen grandes problemas en componentes, tareas o cálculos que puedan resolverse en simultáneo. A continuación, haremos una introducción teórica de los aspectos que son relevantes para luego explicar cómo se aplica el mismo a las redes neuronales y a la visión computacional.

4.1. Master – Worker.

El modelo Master-Worker es un modelo aplicado a la descomposición de dominio, es decir, el domino del problema se divide en subconjuntos de datos y los mismo son asignados a nodos diferentes.**[8]**

El proceso principal denominado Master es el proceso encargado de coordinar todo el tratamiento y procesamiento del problema, para lo que genera muchos subprocesos, que son ejecutados como procesos independientes denominados Workers, y en general se ejecutan en procesadores independientes. La interacción que existe entre ellos es que el Master inicia los procesos Worker, les asigna el trabajo a realizar, y estos devuelven el resultado al proceso Master. En la figura 9 se encuentra graficado este concepto.

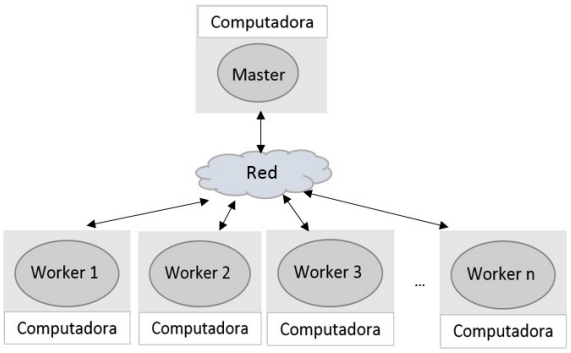

Figura 9:Esquema del modelo Master-Worker.

5. Propuesta de trabajo

En las secciones que siguen se detallará cómo se pretende trabaja con redes neuronales y redes convolucionales, el objetivo de paralelizarlas y lo que se pretende lograr para aportar estos conceptos como ayuda al modelo de predicción de incendios ya existente.

5. 1. Propuesta de trabajo en redes neuronales

Dado que este trabajo está propuesto por una alumna de grado de la carrera de Ingeniería en Sistemas de Información, al principio de esta investigación se contaba con poca experiencia en redes neuronales, y se decidió comenzar a estudiar el perceptrón multicapa. También se tuvo en cuenta que en el LICPaD se trabajaba con lenguaje C/C++ y que una de las pocas redes neuronales que se pueden implementar en este lenguaje es el perceptrón, debido a que es más sencillo de implementar cuando no se cuenta con clases. Luego de determinar que se utilizaría el perceptrón para experimentar, se estudió cómo funcionaba y las formas de pasaje de parámetros, o valores de entrada que podía tener el mismo. Dado que se pretende pasarle una gran cantidad de variables para poder aplicarlo a la predicción de incendios decimos trabajar con el pasaje de entradas por archivo para mantener un buen orden y simpleza en el código.

Como se detalló en las secciones anteriores, una red neuronal está compuesta por una capa de entrada, capas ocultas y una capa de salida. Se propone determinar al Master como el encargado de almacenar datos y de realizar el envío y recepción de estos. Este, además, realizará los cálculos en la capa de entrada, para transformar los datos de entrada en valores utilizables por la red neuronal.

En segundo lugar, proponemos considerar a cada capa oculta y a la capa de salida como instancias donde una o más neuronas, contenidas en las mismas, serán representadas por un proceso. El Master distribuirá la cantidad de neuronas de cada capa -n- por procesador -p-, realizando la división n/p. De esta forma, cada procesador -*Worker*- quedaría a cargo de cierta cantidad de neuronas y realizaría diversos cálculos en cada neurona que le fuera asignada obteniendo una salida.

Luego de ingresar los datos de entrada en la capa de entrada, se obtendría sus salidas que serán enviadas a cada proceso Worker.

El procedimiento explicado anteriormente representa el primer ajuste de pesos y cálculo de salidas de las neuronas de las capas.

El algoritmo de *backpropagation* ajusta los pesos de las entradas de cada neurona hasta que el error de salida, comparado con la salida esperada, sea mínimamente aceptable (como se explica en la Sección 2.4). Por lo tanto, primero se calcula el error de cada salida de las neuronas en la capa de salida y luego, estos se transmitirán a cada procesador. Recordemos que cada proceso se sigue manteniendo a cargo de las neuronas asignadas por el Master al principio del algoritmo.

Una vez recibido los errores, cada neurona procederá a realizar sus ajustes de pesos y el cálculo de su error de salida. Este procedimiento se repetirá en cada capa, excluyendo la capa de entrada.

Finalmente, se realizará nuevamente el cálculo de las salidas con los ajustes de pesos aplicados y en el caso que el error no sea aceptable, volverá a aplicar el algoritmo *backpropagation* cuantas veces sea necesario.

El propósito de lo explicado anteriormente es ver si se puede potenciar el rendimiento del perceptrón para aplicarlo a problemas como sería la predicción de incendios. Las redes neuronales proporcionan herramientas de gran ayuda a la hora de trabajar con predicciones y se pretende aplicarlas en el modelo anteriormente mencionado para tratar la incertidumbre de variables con la que se trabaja.

El objetivo de paralelizar la red neuronal y su aprendizaje es que, al trabajar con una gran cantidad de datos, el análisis de los mismos y el aprendizaje de la red neuronal podrían llevar mucho tiempo, por lo que se pretende potenciar su rendimiento y tiempo paralelizándolo.

5.2. Propuesta de Trabajo con redes convolucionales

La red toma como entrada los pixeles de una imagen. Si tenemos una imagen con apenas 28×28 pixeles de alto y ancho, eso equivale a 784 neuronas. Si tenemos una imagen a color, necesitaríamos 3 canales (red, green, blue) y entonces usaríamos 28x28x3 = 2352 neuronas de entrada. Esa es nuestra capa de entrada.

La propuesta de trabajar con las redes neuronales convolucionales es encontrarle un objetivo práctico a la hora de analizar imágenes o mapas satelitales.

En el modelo de predicción de incendios se trabaja constantemente con mapas, **[9]** al aplicar visión computacional a través de redes convolucionales se busca automatizar y hacer lo más rápido posible este análisis.

La propuesta de paralelización, es similar a la aplicada en las redes neuronales, la descomposición será de dominio, es decir, se repartirá el dominio de problema (datos), en este caso mapas, en los diferentes nodos.

El nodo master se encargaría de tener y repartir las imágenes satelitales, en los diferentes nodos de la red convolucional. Los diferentes nodos tendrán su propia red convolucional y manejarán un gran número de neuronas.

En un futuro se estudiará la descomposición funcional de la misma, teniendo en cuenta las diferentes capas con las que cuenta la red convolucional.
Actualmente se está estudiando la librería OpenCV **[10],** que es una librería de inteligencia artificial con muchas herramientas de visión computacional. La misma nos permite manipular imágenes y videos. También se estudia la posibilidad de desarrollarlas en el lenguaje C, debido a los aspectos anteriormente mencionados.

6. Conclusiones

Este trabajo menciona los conceptos estudiados para entender el funcionamiento de una red neuronal y los conceptos que se aplicarán en ésta para lograr paralelizar su aprendizaje. Actualmente, se sigue estudiando el algoritmo de la construcción y el aprendizaje de una red neuronal para realizar la implementación y la prueba práctica de los conceptos mencionados al paralelizarla.
Se buscará también mejorar la granularidad y la asignación de tareas del programa, es decir, cómo se divide el programa en los procesadores para aumentar la eficiencia del mismo ya que los conceptos presentados darían una mejora lineal. Se estudiarán funciones para el traspaso de mensajes y sincronización de las partes de un programa más eficientes que brinden, al menos, una mejora logarítmica al paralelizar. Esta eficiencia puede mejorar a costa de mayor complejidad para implementarla.
También nos encontramos estudiando cómo llevar a cabo una red convolucional y a partir de esto comenzar a estudiar la posible implementación de las formas de paralelización mencionadas.

7. Referencias

[1] Pedro Ponce Cruz, "Inteligencia Artificial Aplicada a la ingeniería", Alfaomega, 2001.

[2] Germán Bianchini. Paola Caymes Scutari, Miguel Méndez Garabetti, Evolutionary-Statistical System: a Parallel Method for Improving Forest Fire Spread Prediction. Journal of Computational Science (JOCS) Vol 6 pp. 58-66. ISSN: 1877-7503 doi: 10.1016/j.jocs.2014.12.001 Elsevier.

[3] Carlos Alberto Ruiz, "Redes Neuronales: Conceptos básicos y aplicaciones", UTN-Facultad Regional de Rosario, 2005.

[4]http://www.sc.ehu.es/ccwbayes/docencia/mmcc/docs/t8neuronales.pdf, accedida el 20/08/19.

[5] Rivera E., "Entrenamiento de redes neuronales en algoritmos evolutivos", 2005.

[6] Matich, D. J., "Introducción a las Redes Neuronales Artificiales",2001.

[7] Sucar Enrique, "Vision Computacional",

[8] Barry Wilkinson, Michael Allen. "Parallel Programming" (2005). Pearson.

[9] G. Bianchini, P. Caymes Scutari, "Metodos basados en computación de alto rendimiento para predecir el comportamiento de incendios forestales" E-ICES7. ISBN 978-987-1323-27-2. Edit. CNEA. pp. 28-36, 2012.

[10]Convutional Neural Network – La teoría explicada en español. https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/