**Visión computacional. Redes neuronales convolutivas para la clasificación de imágenes (pruebas en la base de datos caltech-101).**

**Autor: Jose Diego Marin Contreras.**

**Introducción.**

La inteligencia artificial (IA) es una rama de la computación encargada de estudiar y diseñar entes artificiales inteligentes que son mayoritariamente programas, como por ejemplo, sistemas capaces de actuar como un humano (Russell y Norvig, 2003, P. 2), que puedan procesar la información que obtienen de su entorno. Un ejemplo de esto es la visión, en este aspecto el equivalente en esa área es la visión artificial, que busca desarrollar técnicas para la adquisición y procesamiento de imágenes digitales, con el fin de extraer y medir determinadas propiedades de las imágenes adquiridas (U.N.Q, 2005, P. 1). Además, un atributo que tiene un agente inteligente es su capacidad de adaptación, ya que en base a la experiencia que adquiere podrá mejorar su desempeño para resolver problemas similares (Díaz, S.F.). El aprendizaje automático o machine learning es una rama de la IA, que estudia el diseño y desarrollo de programas que sean capaces de mejorar con ayuda de experiencias (Alonso y Gutiérrez, S.F.).

**Descripción del problema**

La clasificación de imágenes es una tarea que implica tanto técnicas de visión artificial como de aprendizaje automático. El objetivo de esta tarea es clasificar las imágenes en una serie de clases pre-determinadas; para ello, se dispone de dos conjuntos de imágenes, un conjunto de imágenes que sirven de entrenamiento (fase de aprendizaje), el otro conjunto es para probar la eficiencia del modelo, esta aplicación se usa en diversos procesos científicos y militares, extendiéndose en un amplio rango de sectores industriales para la automatización de tareas anteriormente reservadas para la inspección visual humana, este tipo de tecnología destaca en labores de inspección, debido a que son cuantitativamente más objetivos y consistentes que la inspección humana (Pajares y De la Cruz, 2010, P. 74).

Sin embargo, el reconocimiento visual de objetos es un problema computacional extremadamente complejo. Uno de los mayores problemas es que cada objeto puede modelarse en distintas situaciones, como lo son la posición del objeto, la iluminación y el fondo (el cual es relativo al espectador), por consecuencia hay un enorme número de Imágenes 2-D que representa a un solo objeto, aun así el cerebro humano es capaz de resolver ese problema sin esfuerzo. Para el progreso en el entendimiento de la solución que da el cerebro humano en la tarea de reconocimiento de objetos, se busca construir un sistema artificial que logre emular las habilidades visuales del cerebro humano. Actualmente no está muy claro cómo se debe evaluar el progreso que se tiene para lograr dicha meta, por lo que en la práctica se elige un conjunto de imágenes para probar el rendimiento del sistema y compararlo con las técnicas publicadas en la literatura (Pinto, Cox y DiCarlo, 2008, P. 8).

**Desarrollo.**

**Herramientas.**

Para atender este problema de la clasificación de imágenes existe una gran variedad de técnicas como por ejemplo, arboles de decisión, algoritmos genéticos, máquinas de vectores soporte, etc. Pero en este proyecto se hace uso de las redes neuronales, de acuerdo con Pajares y De la Cruz (2010, P. 99 y 101-102), las redes neuronales son estructuras paralelas inspiradas en la redes neuronales biológicas, por lo que está constituido por una diversidad de elementos que en conjunto pueden dar lugar a un procesamiento no lineal complejo.

Para construir el modelo se utilizó Python, el cual es un lenguaje de programación de alto nivel orientado a objetos, este lenguaje de programación destaca por su gran potencia, su facilidad de aprender la sintaxis y su tipado dinámico, haciendo de este lenguaje el ideal para desarrollar scripting, aplicaciones en diversas áreas y sobre la mayoría de las plataformas (Rossum, 2009, P. 7), y además haremos uso de Keras, que es una librería de alto nivel, usado especialmente para construir redes neuronales. Keras está escrito en el lenguaje Python y es compatible con ambas versiones de Python (2.7 y 3.5), fue especialmente desarrollado para ser rápido en ejecución. Tienen una interfaz modular que hace más fácil crear inclusive modelos de redes neuronales complejos (Shaikh, 2016). Para hacer las pruebas del modelo se hace uso de Caltech-101, una base de datos de imágenes divididas por clases (101 clases y fotos de fondos de Google), cada clase tiene entre 40 a 800 imágenes. Creada en septiembre del 2003 por Fei-Fei, Marco Andreetto y Marc ’Aurelio Ranzato. En promedio el tamaño de las imágenes es de 300x200 pixeles.

Las redes neuronales a pesar de ser útiles en el procesamiento no lineal de la información, son incapaces de procesar directamente con los datos, por lo que se tiene que hacer manualmente la representación a mano de las características de las imágenes, siendo un problema de aprendizaje de interés por muchos años. Recientemente se ha progresado bastante en ese problema, con el desarrollo de “Deep Learning” (aprendizaje profundo). En el contexto de la inteligencia artificial, el Deep learning es un conjunto de algoritmos de tipo aprendizaje automático, por medio de varios niveles las maquinas extraen información para adquirir conocimiento, esto último es una gran ventaja ya que no se requiere de hacer la identificación de características a mano de los datos, y la computadora puede trabajar directamente con los datos (Gómez, 6 de Abril 2016, P. 6 - 7), al combinar el Deep learning y las redes neuronales, nace un nuevo tipo de red, las redes neuronales convolutivas (CNNs).

**CNN.**

Una red neuronal convolucional es un tipo de red neuronal artificial donde las neuronas corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria de un cerebro biológico. Este tipo de red es una variación de un perceptron multicapa, sin embargo, debido a que su aplicación es realizada en matrices bidimensionales, son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes, entre otras aplicaciones (Wikipedia, S.F.). Las CNNs están inspiradas biológicamente en las arquitecturas entrañables que pueden aprender características invariantes, dando un buen rendimiento en la tarea de reconocimiento de imágenes, cada etapa en la redes convolutivas están compuesta por un banco de filtros y capas de agrupación de características (LeCun, Kevukcuohlu y Farabet, P. 1). En la Ilustración 1 se muestra gráficamente lo que es una CNN.

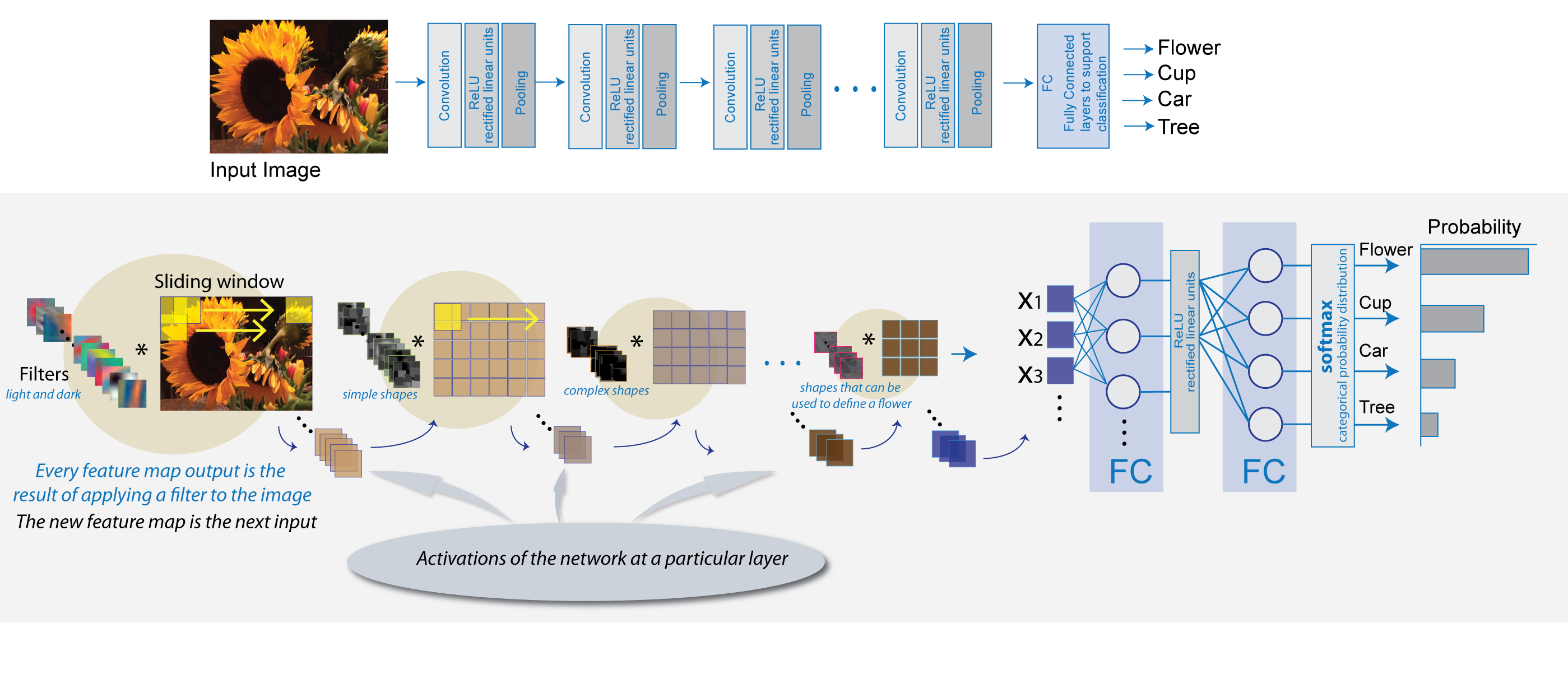


Ilustración 1. Arquitectura de una CNN

Los filtros en procesamiento de imágenes son matrices de MxN que son utilizadas para destacar ciertos elementos con propiedades deseadas de un largo conjunto (Zolzer, 2002, P. 31).

En la Ilustración 2 se muestra como al aplicar filtros en la imagen, se destacan ciertos rasgos distintivos de la imagen, como son la cara de la persona y el sombrero.



Ilustración 2. Imagen sin filtro y después con filtros.

Otro elemento importante en las capas convolutivas es la convolucion, la cual es una operación en el procesamiento de imágenes en la que se puede imaginar a una como la imagen de entrada y (una matriz llamada el núcleo) como un filtro en la imagen, dando como resultado una nueva imagen diferente, en la Ilustración 2 hay un ejemplo de como la convolucion ayuda a destacar las características más importantes del objeto para así hacer una mejor generalización y por consecuencia clasificar posteriores ejemplos. En la Ilustración 3, hay el ejemplo de convolucion.

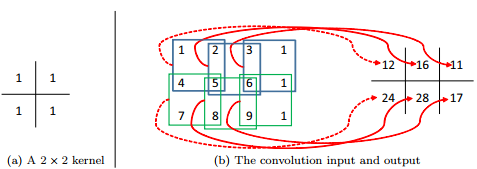


Ilustración 3. Convolucion.

**Procedimiento.**

Para el tamaño de la muestra de las imágenes de entrenamiento y testeo, se siguió el procedimiento de (Fei-fei, Fergus y Perona , 2006), y se seleccionó 30 imágenes por clase para el entrenamiento, para el testeo no se usaron más de 50 imágenes por clase. En la Tabla 1, se describen las características del modelo los filtros, el tamaño de las imágenes, etc.

En la Tabla 2 están descritas las características del equipo de cómputo en el que se aplica las pruebas del modelo.

En la Tabla 3 se encuentran los resultados finales del accuracy que obtuvo mi modelo contra el de otros métodos y modelos usando como prueba la base de datos de Caltech-101 que han publicados.

|  |  |
| --- | --- |
| ***Arquitectura del modelo.*** | |
| **Capas convolutivas.** | |
| **Conv 1.** | + 96 filtros de 7x7.  + Activacion: relu.  + Forma de la entrada: 224x224x3.  + Strides: 2x2.  + MaxPooling: 3x3.  + Strides: 2x2.  + BatchNormalization. |
| **Conv 2.** | + 256 filtros de 5x5.  + Activacion: relu.  + Strides: 2x2.  + MaxPooling: 3x3.  + Strides 2x2.  +BatchNormalization. |
| **Conv 3.** | + 384 filtros de 3x3.  + Activacion: relu.  + Strides: 1x1. |
| **Conv 4.** | + 384 filtros de 3x3.  + Activacion: relu.  + Strides: 1x1. |
| **Conv 5.** | + 256 filtros 3x3.  + Activacion: relu.  + Strides: 1x1.  + MacPooling: 3x3.  + Strides: 2x2. |
| **Capas de la red neuronal.** | |
| **Completamente conectadas.** | + Neuronas: 4096.  + Activacion: relu.  + Dropout: 0.5 |
| **Completamente conectadas.** | + Neuronas: 4096.  + Activacion: relu.  + Dropout: 0.5 |
| **Completamente conectadas.** | + Neuronas: 101.  + Activacion: softmax. |
| **Optimizador.** | |
| **SGD.** | + Learning rate: 0.01.  + Momentum: 0.9.  + Decay: 0.0005.  + Nesterov: True. |
| **Epocas.** | + Batch size = 128.  + Epochs = 512. |

Tabla 1 Arquitectura del modelo

|  |  |
| --- | --- |
| ***Arquitectura del equipo de cómputo.*** | |
| **DELL PRECISION TOWER 5810.** | |
| **Procesador.** | Intel(R) Xeon(R) CPU ES-1620 v3 @ 3.50 GHz. |
| **Memoria (RAM).** | 64.0 GB. |
| **Gráficos.** | NVIDIA Quadro K4200. |
| **Memoria dedicada de gráficos.** | 4096 MB. |

Tabla 2 Especificaciones

**Resultados.**

|  |  |
| --- | --- |
| ***Resultados.*** | |
| **Arquitecturas.** | **Accuracy.** |
| Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition. | 91.44% ± 0.7% |
| Visualizing and Understanding Convolutional Networks. | 86.5% ± 0.5% |
| Group-Sensitive Multiple Kernel Learning for Object Categorization | 84.3% |
| Classification using Intersection Kernel Support Vector Machines is Efficient. | 52% |
| Object Recognition with Features Inspired by Visual Cortex. | 42% |
| Mi arquitectura. | 50.198% |

Tabla 3 Resultados.

En el archivo de Apéndice pruebas, se encuentran todas la pruebas hechas a la base de datos Caltech-101.

**Conclusiones.**

Con los resultados obtenidos a pesar de no superar a los resultados obtenidos en la literatura, se ha demostrado que las redes neuronales convolutivas arrojan relativamente buenos resultados, ya que el modelo tuvo que clasificar entre 101 clases diferentes. Puede que mi arquitectura haya fallado debido a factores tales como:

+ Overfiting. El modelo se ajustaba tan bien a los datos de entrenamiento que al momento de aplicar el conjunto de test la arquitectura le costaba identificar las imágenes de ese conjunto. No generaliza bien el modelo.

+ Mejorar la fase de aprendizaje. Es posible que la capas convoluciones no extraiga las características necesarias para identificar correctamente las clases

+ Distorsión de las imágenes. Los objetos de las imágenes se deforman y ya no se asemejan entre sí, haciendo difícil comparar y generalizar las características.

**Referencias.**

+ Faizah Shaikh. 16 de Octubre del 2016. Tutorial: Optimizing Neural Network using keras (with Image recognition case study). Recuperado de: <https://www.analyticsvidhya.com/blog/2016/10/tutorial-optimizing-neural-networks-using-keras-with-image-recognition-case-study/#one>

+ Fei-fei, L., Fergus, R., y Perona. 2006. P. One-shot learning of object categories.

IEEE Trans. PAMI, 2006

+ Fernando Díaz Gómez. Aprendizaje Automático: método y aplicaciones. Dpto. de Informática. E.U. de Informática – Uva

+ Gonzalo Pajares Martinsanz y Jesus Manuel de la Cruz Garcia. 2010. Aprendizaje Automático Un enfoque Práctico. Ra-Ma.

+ Guido Van Rossum. Septiembre del 2009. El tutorial de Python.

+ José A. Alonso Jiménez y Miguel A. Gutiérrez Naranjo. Tema 10: Introducción al

Aprendizaje Automático. Dpto. de Ciencias de la computación e Inteligencia

Artificial. Universidad de Sevilla.

+ Nicolas Pinto, David D. Cox y James J. DiCarlo. 2008. Why is Real-World Visual Object Recognition Hard?. Digital Access to scholarship at Harvard.

+ Stuart J. Russell y Peter Norvig. 2003. INTELIGENCIA ARTIFICIAL UN ENFOQUE MODERNO. Segunda Edicion. Pretince Hall.

+ U.N.Q. (Universidad Nacional de Quilmes). Octubre de 2005. Aspectos de un Sistema de Visión Artificial.

+ Wikipedia. Redes neuronales Convolucionales. Recuperado de: <https://es.wikipedia.org/wiki/Redes_neuronales_convolucionales>

+ Yann LeCun, Koray Kavukcuoglu y Clément Farabet. Convolutional Networks

and Applications in Vision. Computer Science Department , Courant Institute of

Mathematical Sciences, New York University.

+ Udo Zolzer. 2002. Chapter 2. Filters.DAFX: Digital Audio Effects.