

# UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN.

## Facultad de Ingeniería Mecánica y Eléctrica.



UANL

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN.



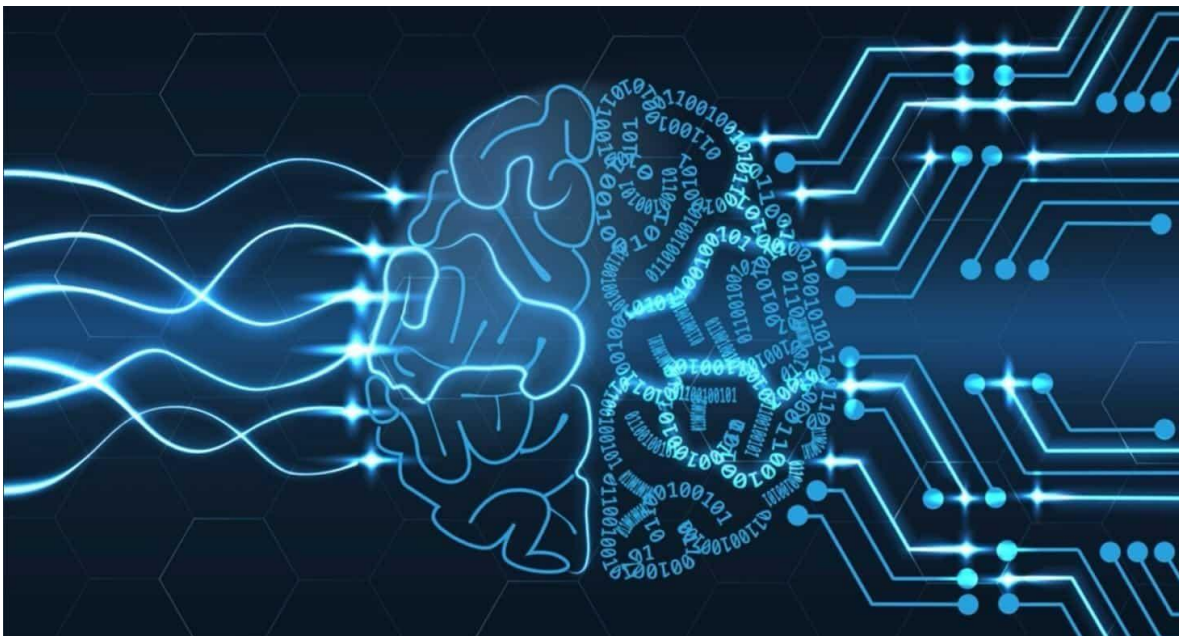
FIME

FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

---

### *Entrenamiento de Red Neuronal Artificial simple*

---



### **Autores**

NESTOR OCTAVIO GUAJARDO RODRIGUEZ

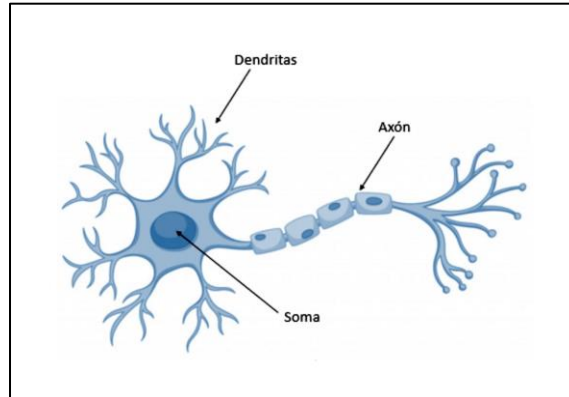
JAZIEL VILLARREAL DUARTE

SERGIO MAXIMINO MELCHOR VILLEGAS

KEVIN ALEXIS MARTINEZ GARCIA

## Marco Teórico

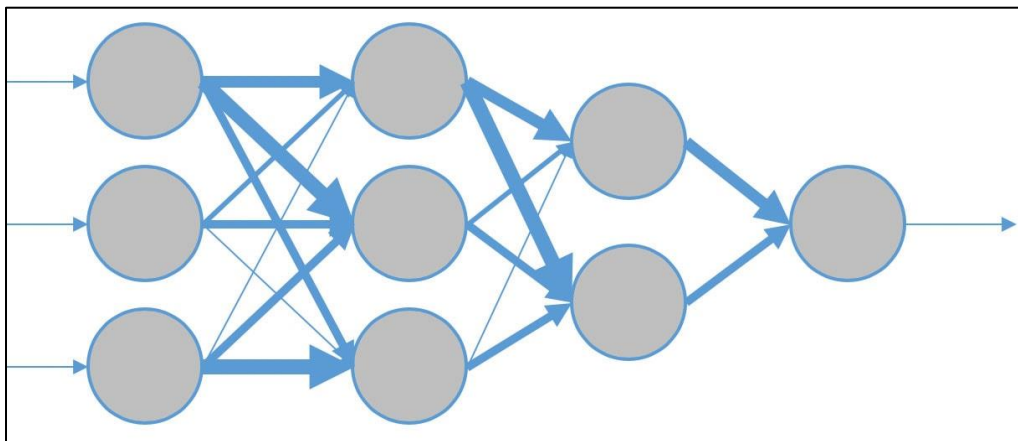
Las redes neuronales artificiales están basadas en el funcionamiento de las redes de neuronas biológicas. Las neuronas que todos tenemos en nuestro cerebro están compuestas de dendritas, el soma y el axón: Las dendritas se encargan de captar los impulsos nerviosos que emiten otras neuronas. Estos impulsos, se procesan en el soma y se transmiten a través del axón que emite un impulso nervioso hacia las neuronas contiguas.



En el caso de las neuronas artificiales, la suma de las entradas multiplicadas por sus pesos asociados determina el “impulso nervioso” que recibe la neurona. Este valor, se procesa en el interior de la célula mediante una función de activación que devuelve un valor que se envía como salida de la neurona.

Del mismo modo que nuestro cerebro está compuesto por neuronas interconectadas entre sí, una red neuronal artificial está formada por neuronas artificiales conectadas entre sí y agrupadas en diferentes niveles que denominamos capas:

Una capa es un conjunto de neuronas cuyas entradas provienen de una capa anterior (o de los datos de entrada en el caso de la primera capa) y cuyas salidas son la entrada de una capa posterior.



Las neuronas de la primera capa reciben como entrada los datos reales que alimentan a la red neuronal. Es por eso por lo que la primera capa se conoce como capa de entrada. La salida de la última capa es el resultado visible de la red, por lo que la última capa se conoce como la capa de salida. Las capas que se sitúan entre la capa de entrada y la capa de salida se conocen como capas ocultas ya que desconocemos tanto los valores de entrada como los de salida.

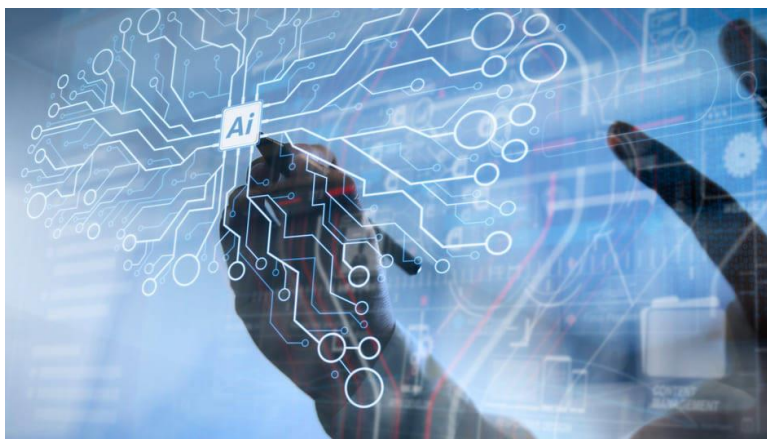
Una red neuronal, por lo tanto, siempre está compuesta por una capa de entrada, una capa de salida (si solo hay una capa en la red neuronal, la capa de entrada coincide con la capa de salida) y puede contener 0 o más capas ocultas. El concepto de Deep Learning nace a raíz de utilizar un gran número de capas ocultas en las redes.

### **¿Cómo se entrena una red neuronal?**

Entrenar una red neuronal consiste en ajustar cada uno de los pesos de las entradas de todas las neuronas que forman parte de la red neuronal, para que las respuestas de la capa de salida se ajusten lo más posible a los datos que conocemos.

En la siguiente animación, podemos ver un ejemplo muy simplificado del proceso de entrenamiento de una red para la detección de, en este caso, un gato en una imagen. El grosor de cada flecha representa el peso que tiene esa entrada en la red neuronal. Como podemos comprobar, a medida que se van ajustando los pesos el error disminuye.

Si queremos conseguir que la red neuronal sea capaz de generalizar e identificar gatos en cualquier imagen, es importante utilizar un elevado número de imágenes para realizar el entrenamiento, tanto de imágenes que son gatos (etiquetadas como 1) como de imágenes que no son gatos (etiquetadas como 0), incluyendo la mayor variabilidad posible. Con esto, la red será capaz de ajustar sus parámetros para satisfacer en la medida de lo posible todas las imágenes, por lo que será capaz de extraer de manera precisa las características que identifican la presencia de un gato en una imagen.



### **Tipos de redes neuronales**

Las redes neuronales se pueden clasificar en base a su estructura:

- **Monocapa:** la red neuronal está formada por una única capa de neuronas que recibe valores del exterior de la red, los procesa y genera unos valores de salida que entrega como resultado. Las neuronas de la capa no se interconectan entre ellas.
- **Multicapa:** es el tipo de red neuronal más extendido. Consta de una capa de entrada que recibe valores del exterior, una serie de capas intermedias u ocultas que van procesando la información y una capa de salida que entrega los resultados de la red neuronal. Las neuronas de una capa se interconectan con todas las neuronas de la capa siguiente, pero no entre ellas.
- **Convolutacional:** son similares a las redes neuronales multicapa, pero la diferencia radica en que en estas las neuronas de cada capa no se interconectan con todas las de la capa siguiente, si no con un subconjunto de estas, y no se interconectan entre las neuronas de la misma capa. En este caso se habla de especialización de las neuronas (grupos) y se reduce la cantidad de unidades necesarias y la complejidad de los sistemas que utilizan este tipo de redes.
- **Concurrentes:** en estas redes las neuronas no se organizan en capas, sino que las neuronas están interconectadas entre ellas de manera no estructurada en capas. Esto permite que estas redes tengan memoria, es decir, que la información generada en iteraciones anteriores afecte al resultado del procesamiento en un tiempo futuro.
- **Radiales, etc.**

### Aplicaciones de las redes neuronales

Los principales ejemplos, o al menos los más conocidos, de la utilización práctica de las redes neuronales son:

- Sistemas de reconocimiento de voz como los empleados en los altavoces inteligentes o HomePod de Amazon, Google o Apple.
- Vehículos de conducción autónoma de Tesla y Uber.
- Los chatbots como Siri de Apple, Alexa de Amazon y Cortana de Microsoft.
- Sistemas de seguridad perimetral para la detección de intrusiones en tiempo real mediante el procesamiento de imagen captada por las cámaras de videovigilancia.

## Metodología

En cuanto a lo primero que realizamos es el material que utilizaremos para poder entrenar las redes, en este caso vamos a utilizar los números hechos por nosotros, estos números los hicimos desde nuestro celular, hicimos los 400 del cero al 9 cada integrante, una vez teniendo estos números los subimos como carpeta a Google colab, que es donde vamos a programar todo para poder modificar la resolución y colores, programamos el formato de los números dejándolos con una resolución de 28x28 MP una vez teniendo esa medida en todos los números, colocamos etiquetas para que se vean reflejados con ayuda de un **len**.

Primero agregamos los imports generales y después los imports específicos para Tensorflow y los que vamos a utilizar. Comenzamos con el código interesante; obtenemos el set de datos que utilizaremos en el set de la librería, obtenemos 80% de los datos de entrenamiento como training datos y los 20% para evaluación en tests de datos. Definimos etiquetas simples de texto para cada posible respuesta en nuestra red.

Obtenemos la cantidad de ejemplos en variables para utilizarlo después pondremos los datos de entrenamiento y de evaluación. En este punto definiremos una función de normalización; lo único que hay que hacer es que el rango de valores de los pixeles sea de 0 a 1 en lugar de 0 hasta 255 entonces solo dividimos el numero entre 255. Llevamos a la función de normalización en cada dato de ambos setes de datos.

Ahora definiremos la estructura de la red e indicamos primero la capa de entrada con 784 neuronas especificando que llegará en una forma cuadrada de 28 por 28 pixeles, agregamos dos capas ocultas densas y con 64 neuronas cada una y finalmente la capa de salida.

Usaremos reglo para las capas ocultas y softmax es requerida en la capa de salida para las redes de clasificación, compilemos el modelo especificando la función de costo a utilizar y optimizador. Por último, indicamos que queremos poder obtener para imprimir en consola el dato de la precisión de nuestra red; prepararemos un poco los setes de datos que utilizaremos, especificamos un tamaño de lote de 32. Los datos de entrenamiento los ordenamos de manera aleatoria y en ambos especificamos el tamaño del lote.

Ahora lo bueno: el entrenamiento. Especificamos los datos de entrenamiento cuantas épocas o vueltas completas a todos los datos del test se usarán en entrenamiento. Ya que la red fue entrenada usaremos los datos de evaluación para ver que tan bien entrenada quedó. Imprimamos los resultados de la precisión. Ahora al ejecutarlo comienza las cinco épocas de entrenamiento y al terminar podremos ver la precisión que tuvo la red.

## Resultados

- Mostrar el cambio de tamaño de las imágenes de dígitos creados por ustedes (28 x 28 pixeles).

<https://colab.research.google.com/drive/10hozzZ6sTyk-shKx084aAA7LbIsqYSOh?usp=sharing>

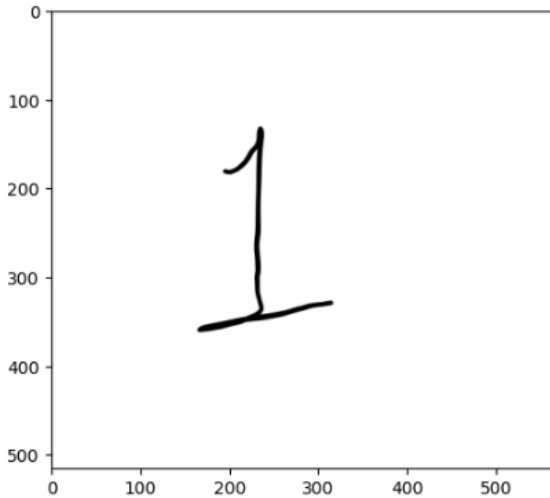
- Enlace del repositorio en donde se encuentra la red neuronal

[https://github.com/Kevin8798/IA-y-RN-Grupo-002/blob/main/Proyecto\\_final\\_Equipo5/Proyecto\\_final\\_Equipo5.ipynb](https://github.com/Kevin8798/IA-y-RN-Grupo-002/blob/main/Proyecto_final_Equipo5/Proyecto_final_Equipo5.ipynb)

```
[207] x = "/content/1 - copia (2).png"
      test_image = cv2.imread(x, cv2.IMREAD_GRAYSCALE )
```

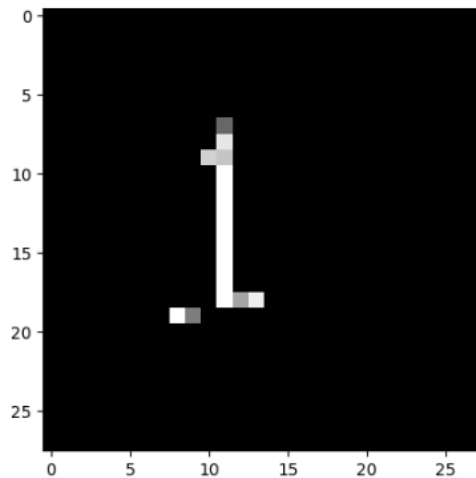
```
plt.imshow(test_image, cmap="gray")
```

<matplotlib.image.AxesImage at 0x7f4f3d464130>



```
x = "/content/1 - copia (2).png"
test_image = cv2.imread(x, cv2.IMREAD_GRAYSCALE )
plt.imshow(test_image, cmap="gray")
img_resized = cv2.resize(test_image, (28,28), interpolation = cv2.INTER_LINEAR)
img_resized = cv2.bitwise_not(img_resized)
plt.imshow(img_resized, cmap="gray")
```

<matplotlib.image.AxesImage at 0x7f4f3cb97760>



```
✓ [235] ds_estudiantes.append(img_resized)
      ds_estudiantes_labels.append("9")
```

```
✓ [236] len(ds_estudiantes)
```

10

```
✓ [237] ds_estudiantes_labels
```

['0', '1', '2', '3', '4', '5', '6', '7', '8', '9']



## Discusión

Por lo que la red neuronal funciona de forma precisa, por lo que su único aspecto en contra son los datos por lo que unos de los retos que se presentan en el aprendizaje automático está presente ya que en si el problema radica en sus datos como ya se mencionó anteriormente. Y es que este problema al principio uno pudiese creer que es fácil de evitar pero llevándolo a la práctica es cuando entra la complejidad porque cuál debería ser el comportamiento exacto que debería presentar la red neural. Naturalmente el fin de esta es que replique un comportamiento humano, sí pero hasta qué punto? Por lo que en este proyecto se buscó que esta realizara una predicción de forma certera, pero una vez es aquí donde entra la discusión ya que los retos presentes del aprendizaje automático como los son el sobreajuste (overfitting) y la representatividad de los datos.

Es un hecho que es difícil replicar un comportamiento humano de forma artificial, no obstante esto no justifica que el su resultado también puede llegar a ser artificial y esta división entre lo real y artificial lo que hace que se presenten los datos y resultados al momento de poder trabajar con nuestra red neuronal y con la predicción que esta realizó. Sin embargo podemos proponer un argumento ante el reto del sobre ajuste de datos y es el hecho de la exactitud de predicción que nos puede dar esta, y es que sí porcentaje que nos da de es alto entonces pudieramos decir que sí no se encuentra algún problema con los datos nuestra red neuronal es eficaz y pudiese representar a un experto en la materia la cual esté realizando el trabajo en la vida real, trasladandolo de lo artificial a lo real. Y para fin creemos que es el fin del aprendizaje automático al momento de trabajar con redes neuronales el cual es volver expertos en un trabajo a estas redes mediante buenos entrenamientos y buenos datos natural y artificialmente

## **Conclusiones**

### **Néstor Octavio Guajardo Rodriguez**

En este trabajo honestamente me quedaron muchos aprendizajes, desde lo educativo, hasta lo personal, principalmente no me quedo conforme con el resultado de este trabajo, no representa para nada mis capacidades, el no organizarme me costo caro y este trabajo me lo ha demostrado, lo deje para el final y no pude realizar al 100% lo que se pedía como PIA, en cuanto a estar investigando todo para poder realizarlo, me deja claro que esto es parte de la modernidad, considero que puede servirnos para muchas cosas, reconocer personas, animales, objetos, entre mas cosas, el realizar este trabajo es muy entretenido, mas que nada porque me hace abrir la imaginación y ver más allá, me hace pensar en donde podría meter este sistema y que usos le podría dar, soy de esas personas que hace las cosas con mucha pasión, siempre intento mantener mi ideología de si hare algo, lo hare bien y siempre pienso en ayudar a las demás personas, por lo tanto como ingeniero pienso en cosas que podrían ayudar a muchas personas, cosas que puedan desarrollarme a mi mismo principalmente como persona y poder ayudar más.

### **Sergio Máximo Melchor Villegas**

Lo que aprendí durante este trabajo y a lo largo del curso fue que es complejo el replicar la inteligencia artificial y obvio siempre se tiene que empezar desde lo más básico y fundamental como lo es una neurona. Es gracias a todo el trabajo e información vista a lo largo del curso que pude entender el cómo funciona una red neuronal en el área del aprendizaje automática ya que los resultados obtenidos en esta durante el entrenamiento y la prueba realizada, hace que los retos que están presentes en el Machine Learning sean más comprensibles y se trate de buscar una forma de como poder afrontarlos ya que la respuesta que aprendí es mediante más entrenamiento y mejores datos y es que es así como en la vida real.

### **Kevin Alexis Martínez García**

Nuestro equipo el cual está integrado por 4 integrantes se dio a la tarea de realizar como objetivo una red neuronal que está diseñada para trabajar con imágenes, asignándole importancias a ciertos elementos en la imagen para así poder diferenciar uno de otros. La tarea no era fácil, ya que debíamos de trabajar con algunos programas para así dar a realizar el proyecto asignado, crear una red neuronal lleva su dedicación, en el transcurso tuvimos ciertos errores en la elaboración, a veces nuestros códigos no corrían y no entendíamos el porqué, pero



después de analizarlo detenidamente pudimos dar solución y así terminar, hubo diversas dudas pero pudimos apoyarnos con internet para solucionar estas dudas. Entre todo nuestro equipo nos apoyamos para completar esta tarea, nos quedó mucho aprendizaje y también nos sirvió como reforzamiento para poder aprender un poco más en cuanto a programas y trabajar con códigos.

### **Jaziel**

Como conclusión de esta actividad es que aprendimos a entrenar una red neuronal artificial simple con la ayuda de colab para programar en python para entrenamiento y prueba de las imágenes que nosotros hicimos para la actividad, predecir en las imágenes en que categoría entran y determinar el grado de acierto es decir, en cuantos estuvo correcto las predicciones en nuestro programa de entrenamiento de una red neuronal.

## **Referencias**

Real Python. (2021, 13 octubre). Using the len() Function in Python.  
<https://realpython.com/len-python-function/>

Team, K. (s. f.-c). Keras documentation: MNIST digits classification dataset.  
<https://keras.io/api/datasets/mnist/>

Geron, A. (2019). "Hands-On machine learning with Scikit-Learn, Keras, and Tensorflow

"Concepts, tools and techniques to build intelligent systems" O'Reilly  
Media Chapters 1 & 10

<https://www.xeridia.com/blog/redes-neuronales-artificiales-que-son-y-como-se-entrenan-parte-i>

<https://www.redalyc.org/journal/5537/553768213002/html/>