

KERAS

Julio 2021



DataSphere
Let's generate value

Contenido



REDES NEURONALES.



KERAS



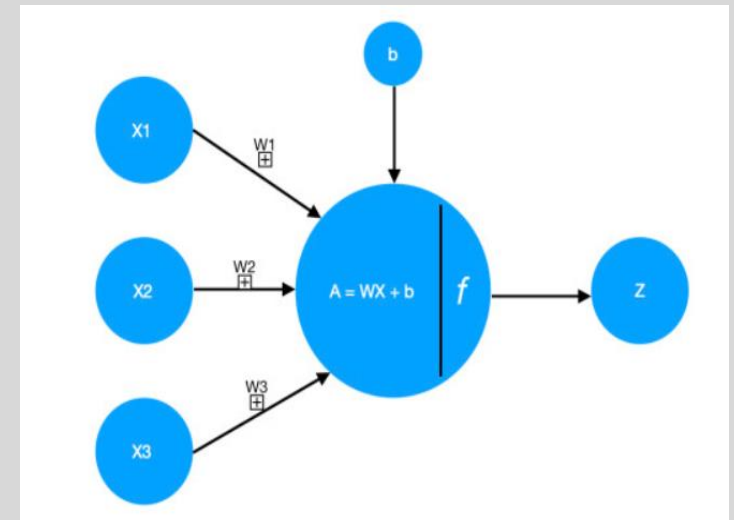
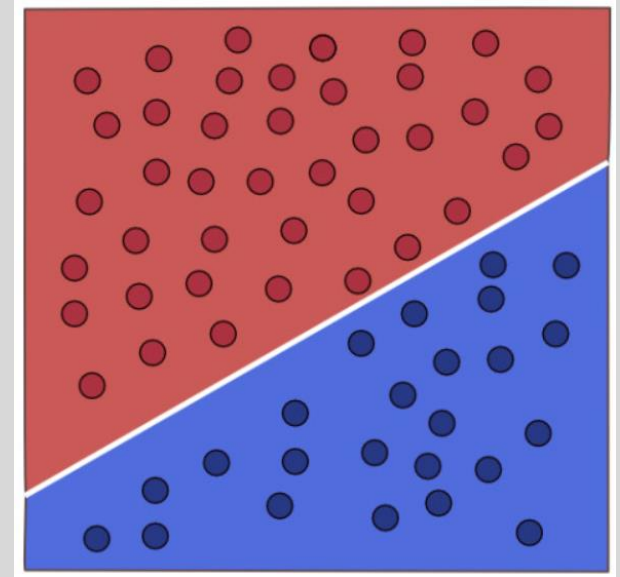
TENSORFLOW.

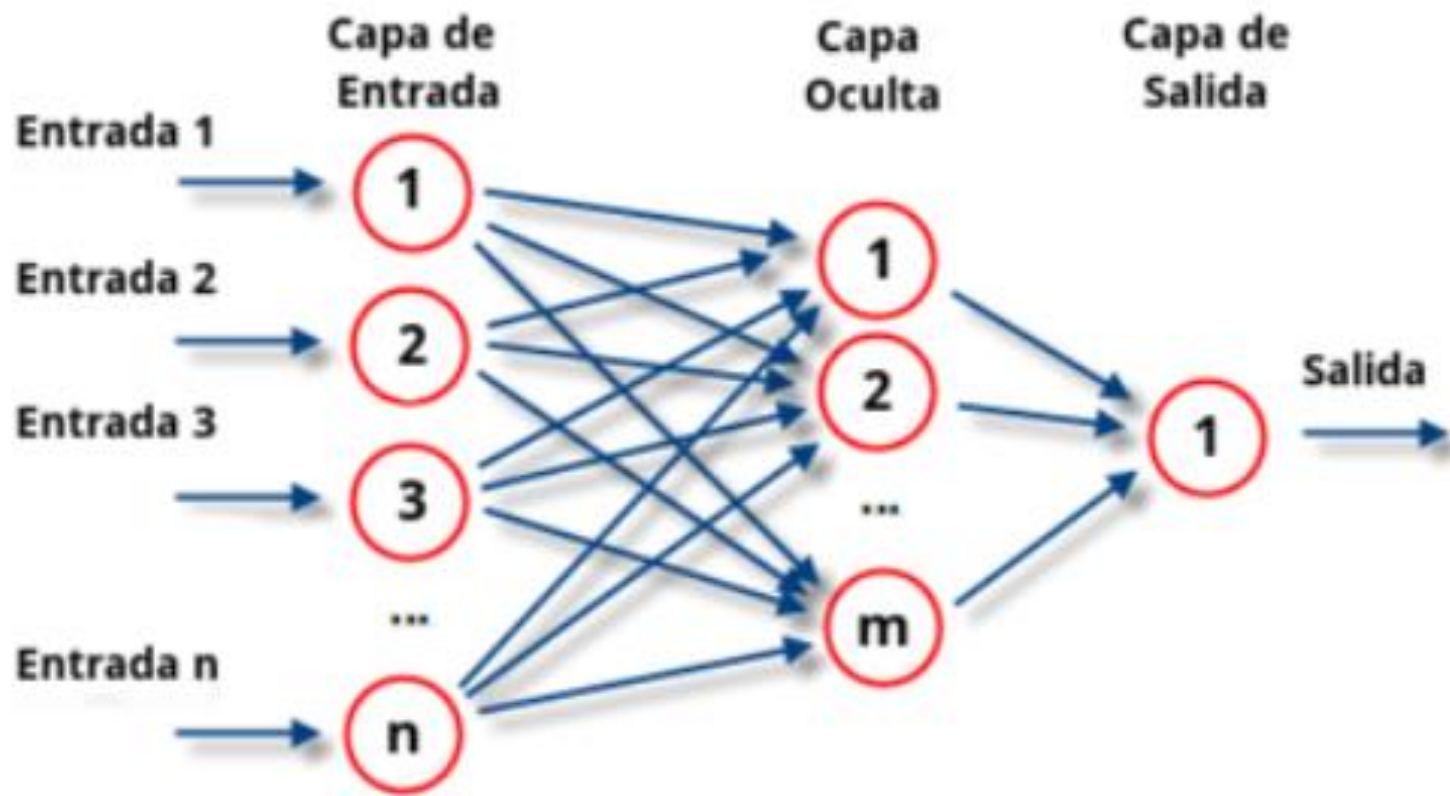
Que es una Red Neuronal

- Las redes neuronales son **sistemas computacionales, inspirados en las neuronas** que constituyen el cerebro de los animales, dotando a los ordenadores de inteligencia artificial. Están formadas por **unidades básicas llamadas neuronas que se conectan entre sí** formando la red neuronal. El objetivo de estos algoritmos es entender datos del mundo real (imágenes, texto, voz..etc), procesarlos y clasificarlos o etiquetarlos.
- Existen desde mediados del siglo pasado
- Se trata de una familia de algoritmos muy potentes con los que podemos modelar comportamientos inteligentes

NEURONA

- Una neurona o nodo es una parte importante de las redes neuronales y pueden haber muchas de ellas repartidas entre varias capas (layers).
- Cada neurona contiene la función
 - $Z = WX + b$ (W es peso, b es un parametro que es igual en todas las neuronas de la misma capa y sirve para controlar el sesgo)
 - Algunos autores representan esos pesos como switches que se pueden regular (potenciometros)
- El resultado de esta función es enviado a la función de activación:
 - $A = \text{sigmoid}(Z)$
 - $A = \text{relu}(Z)$
- y el resultado final (**A**) es enviado a las neuronas de la siguiente capa(layer) o a la función de perdida.



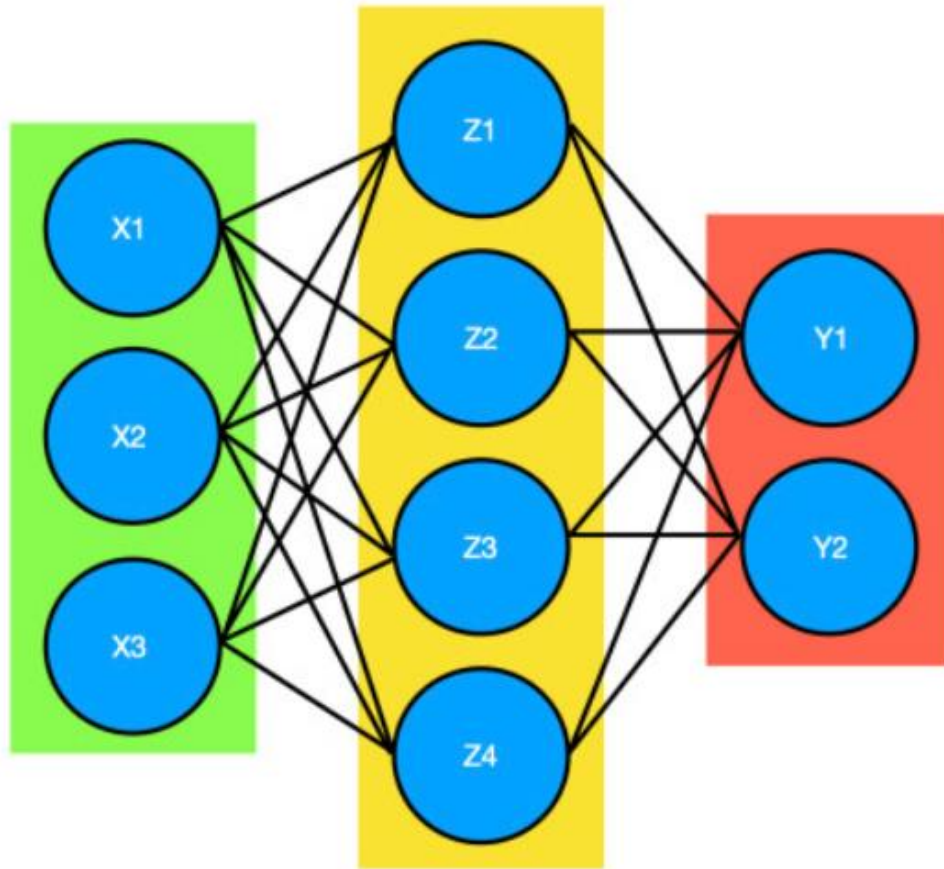


Esquema de una red neuronal

Que es una Red Neuronal

Las redes neuronales son modelos simples del funcionamiento del sistema nervioso. Las unidades básicas son las **neuronas**, que generalmente se organizan en **capas**

Capas en una Red Neuronal



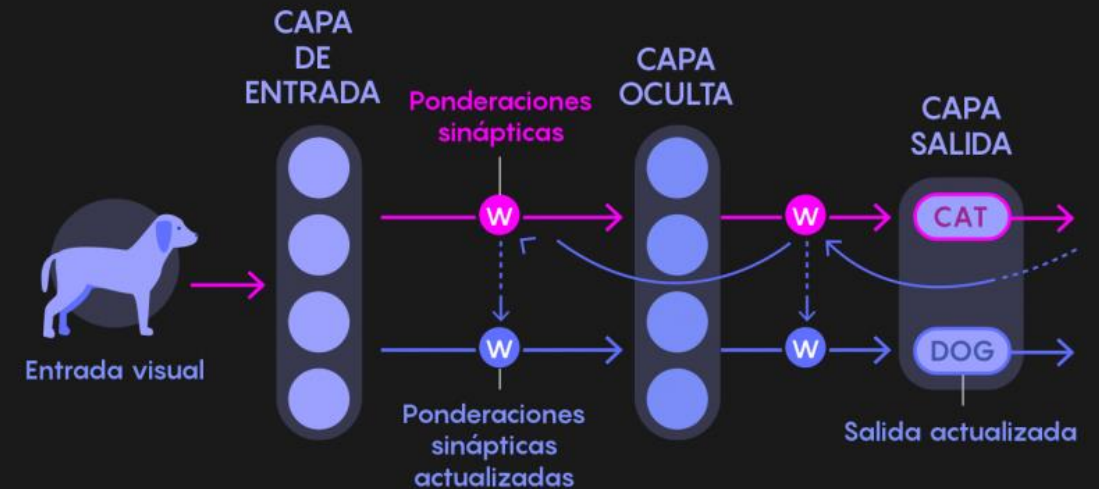
- Hay tres partes normalmente en una red neuronal : una **capa de entrada**, con unidades que representan los campos de entrada; una o varias **capas ocultas**; y una **capa de salida**, con una unidad o unidades que representa el campo o los campos de destino.
- **Capa de entrada (Input Layer):** Esta capa contiene neuronas que representan los datos que la red neuronal usara para entrenar. El numero de neuronas de esta capa depende del numero de características que tengan los datos.
- **Capa oculta (Hidden Layer):** Una red neuronal puede tener varias capas de este tipo, cada una de estas capas contiene neuronas, en una red neuronal tradicional cada una de las neuronas de una capa estan conectadas con todas las neuronas de la siguiente capa.
- **Capa de salida (Output Layer):** Esta capa es la que se encarga de entregar los resultados, si estamos resolviendo un problema de clasificación esta capa tendra un numero de neuronas igual al numero de clases que existan en los datos. El resultado es una lista de probabilidades para cada clase.

OTROS CONCEPTOS

- **Backpropagation**
- Es importante conocer este concepto a detalle, generalmente se conoce el algoritmo de backpropagation como el encargado de optimizar la función de pérdida para mejorar las predicciones de una red neuronal. Este algoritmo se encarga de calcular las derivadas (o gradientes) de los parámetros \mathbf{W} y \mathbf{b} para saber como estos parámetros afectan al resultado de la función de pérdida, esta es una definición que puede ser usada para explicar el algoritmo, pero para ser más precisos, la optimización de una red neuronal se divide en dos partes, la primera es el algoritmo de backpropagation, como mencione este algoritmo se encarga de ver como los valores de \mathbf{W} y de \mathbf{b} afectan al resultado de la función de pérdida y la segunda parte es el algoritmo de optimización, este se encarga de optimizar la red neuronal y cambiar los valores de \mathbf{W} y de \mathbf{b} conforme pasan los ciclos (o epochs). Existen diferentes algoritmos de optimización, unos son mejores que otros, aunque depende del tipo de problema se este resolviendo. Este algoritmo es un parametro de la red neuronal llamado **optimizer**.

Retropropagación

En el algoritmo de la retropropagación, unas conexiones ponderadas entre las neuronas conducen de una entrada a una salida inferida. La información retrocede entonces a través de la red para corregir las ponderaciones sinápticas de las capas ocultas.



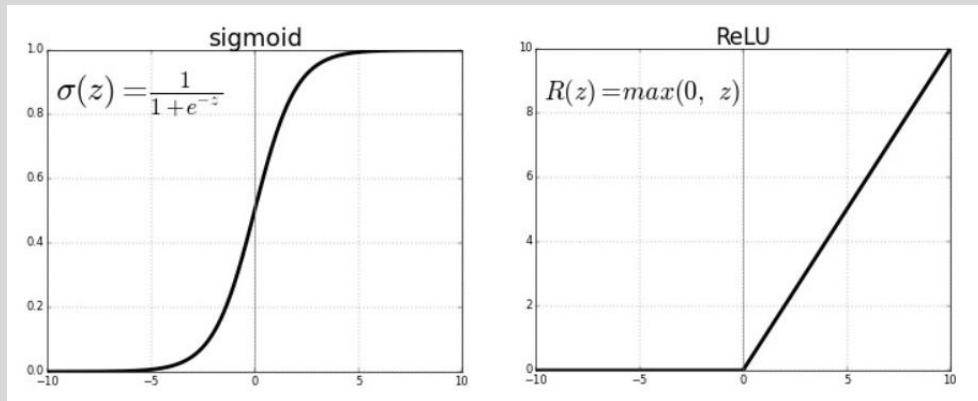
OTROS CONCEPTOS

- **Forwardpropagation**
- Es la manera en la cual las redes neuronales crean las predicciones. En un principio la red neuronal tiene valores de **W** y **b** aleatorios en cada neurona, los datos de entrenamiento pasan por estas neuronas hasta llegar a la capa de salida, en esta capa la red neuronal predice la clase a la cual pertenecen los datos de entrenamiento, estas predicciones las usa la función de pérdida para medir que tan buena es la red neuronal, este ciclo se repite varias veces segun indiquemos y en cada ciclo se ejecuta el algoritmo de backpropagation para actualizar los valores de **W** y **b**.

OTROS CONCEPTOS

- **Epoch**
- Este es el numero de veces que se ejecutaran los algoritmos de forwardpropagation y backpropagation. En cada ciclo (epoch) todos los datos de entrenamiento pasan por la red neuronal para que esta aprenda sobre ellos, si existen 10 ciclos y 1000 datos, cada ciclo los 1000 datos pasaran por la red neuronal. Si se especifica el parametro **batch size** cada ciclo (epoch) tendra más ejecuciones internas, estas ejecuciones se llaman iteraciones, si tenemos un **batch size** de 100, se tendran 10 iteraciones para completar un ciclo, en cada iteración se ejecutan los algoritmos de forwardpropagation y backpropagation, de esta manera la red neuronal actualiza más veces los parametros **W** y **b**. Esta variable también es un hiperparametro.

OTROS CONCEPTOS



- **Activation function**

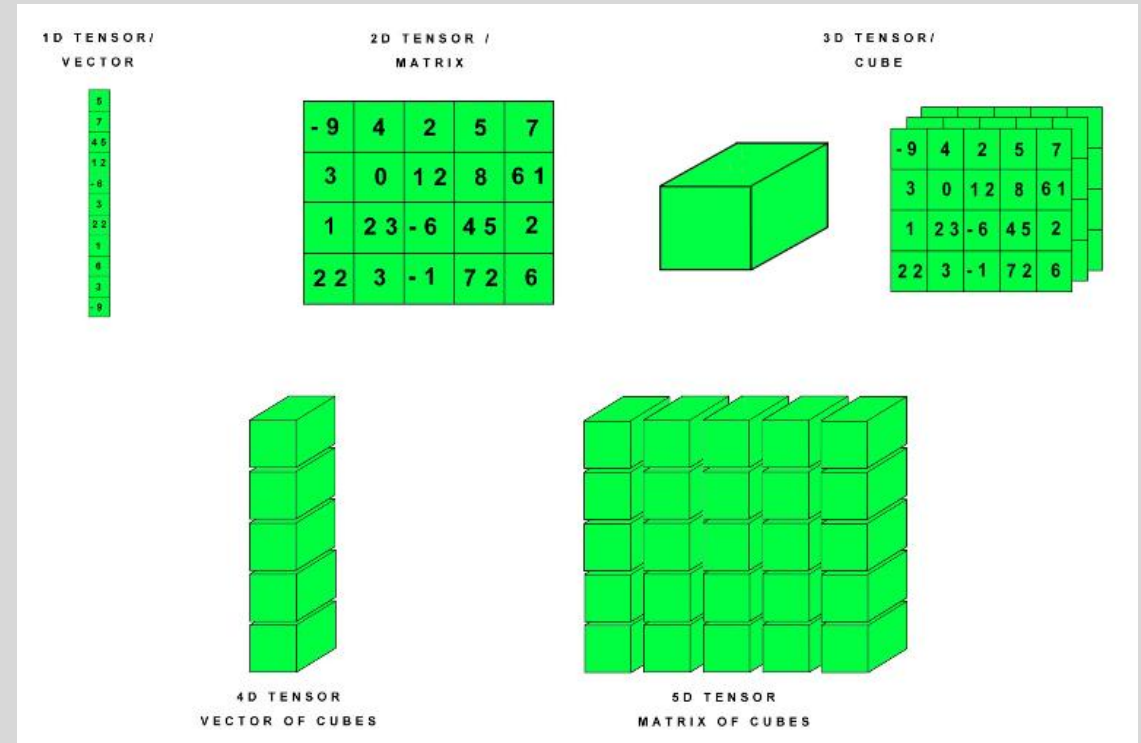
Las funciones de activación se encuentran en cada neurona de una red neuronal y la utilidad más importante que tienen es indicar cuando una neurona se activa o se apaga, dependiendo de la función de activación que se use, esta tendrá ciertos límites, recordemos que primero se calcula la función

$Z = WX + b$ y el resultado de esta función se le pasa a la función de activación, esta última busca si los datos tienen los patrones que busca la neurona o no los tiene.

Las funciones de activación más usadas son sigmoid y relu

OTROS CONCEPTOS

- Los **tensores** son objetos matemáticos que almacenan valores numéricos y que pueden tener distintas dimensiones. Así, por ejemplo, un **tensor** de 1D es un vector, de 2D una matriz, de 3D un cubo



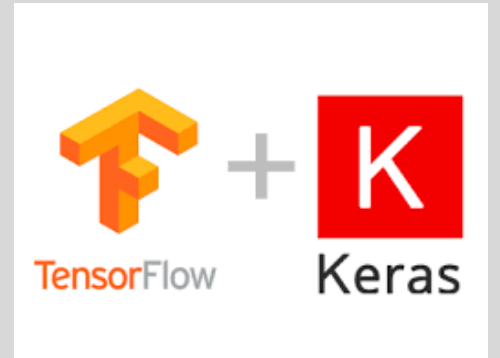
Que es KERAS?

- **Keras** es un framework de alto nivel para el aprendizaje, escrito en **Python** y capaz de correr sobre los frameworks **TensorFlow**, **CNTK**, o **Theano**. Fue desarrollado con el objeto de facilitar un proceso de experimentación rápida. Lo que haremos en este experimento es entrenar modelos de clasificación de imágenes. Esto consiste en dada una serie de imágenes etiquetadas, reconocer una imagen y asignarle dicha etiqueta (por ejemplo, si es la foto de un gato, el modelo reconocerá que hay un gato). Inicialmente fue desarrollada como parte de los esfuerzos de investigación del proyecto [ONEIROS](#) (Open-ended Neuro-Electronic Intelligent Robot Operating System)
- Su autor principal y mantenedor ha sido el ingeniero de Google [François Chollet](#).
- En 2017, el equipo de TensorFlow de Google decidió ofrecer soporte a Keras en la biblioteca de core de TensorFlow.

<https://es.wikipedia.org/wiki/Keras>



Como funciona KERAS



- Keras es una biblioteca que funciona a nivel de modelo: **proporciona bloques modulares** sobre los que se pueden desarrollar modelos complejos de aprendizaje profundo. A diferencia de los *frameworks*, este software de código abierto no se utiliza para operaciones sencillas de bajo nivel, sino que utiliza las bibliotecas de los *frameworks* de aprendizaje automático vinculadas, que en cierto modo actúan como un motor de *backend* para Keras. Las capas de la red neuronal que se quieren configurar se relacionan entre sí de acuerdo con el **principio modular**, sin que el usuario de Keras tenga que comprender o controlar directamente el propio *backend* del *framework* elegido.
- Keras se basa especialmente en las herramientas TensorFlow, Theano y Microsoft Cognitive Toolkit, para las cuales existen **interfaces listas para usar** que permiten un **acceso rápido e intuitivo al *backend* correspondiente**.

Ventajas de KERAS

- **Amplia compatibilidad entre plataformas para los modelos desarrollados:** los modelos desarrollados con Keras son especialmente fáciles de utilizar en diferentes plataformas.
- **Compatibilidad con múltiples motores de *backend*:** Keras no solo te da mucha libertad a la hora de elegir el *backend*, sino que también te permite combinar varios motores.
- **Excelente soporte para múltiples GPU:** con Keras, los recursos necesarios para desarrollar los procesos de aprendizaje profundo se pueden distribuir fácilmente en varios chips o tarjetas gráficas.
- **Desarrollo por parte de grandes empresas:** el mantenimiento y el desarrollo de Keras cuentan con el apoyo de las empresas más importantes del sector. Google, Amazon AWS, Microsoft, Apple y Nvidia, entre otras, están implicadas en el proyecto.

Que es TensorFlow



- De entre las muchas bibliotecas disponibles **la reina indiscutible es TensorFlow**, que se ha impuesto como la librería más popular en Deep Learning. Actualmente, sería difícil imaginar abordar un proyecto de aprendizaje sin ella.
- TensorFlow es una biblioteca desarrollada por Google Brain para sus aplicaciones de aprendizaje automático y las redes neuronales profundas, liberado como software de código abierto en 9 de noviembre del 2015.
- TensorFlow es una librería de computación matemática, que **ejecuta de forma rápida y eficiente gráficos de flujo**. Un gráfico de flujo está formado por operaciones matemáticas representadas sobre nodos, y cuya entrada y salida es un vector multidimensional (o tensor) de datos.
- Las redes neurales son un tipo particular de gráfico de flujo de datos. Por tanto, **TensorFlow y Keras combinan perfectamente** haciendo un tandem que auna potencia, sencillez de uso, y rapidez de ejecución.

Keras VS Tensorflow



Keras	TensorFlow
Keras is a high-level API which is running on top of TensorFlow, CNTK, and Theano.	TensorFlow is a framework that offers both high and low-level APIs.
Keras is easy to use if you know the Python language.	You need to learn the syntax of using various Tensorflow function.
Perfect for quick implementations.	Ideal for Deep learning research, complex networks.
Uses another API debug tool such as TFDBG.	You can use Tensor board visualization tools for debugging.
It started by François Chollet from a project and developed by a group of people.	It was developed by the Google Brain team.
Written in Python, a wrapper for Theano, TensorFlow, and CNTK	Written mostly in C++, CUDA, and Python.
Keras has a simple architecture that is readable and concise.	Tensorflow is not very easy to use.
In the Keras framework, there is a very less frequent need to debug simple networks.	It is quite challenging to perform debugging in TensorFlow.
Keras is usually used for small datasets.	TensorFlow used for high-performance models and large datasets.
Community support is minimal.	It is backed by a large community of tech companies.
It can be used for low-performance models.	It is use for high-performance models.

Referencias

- <https://vincentblog.xyz/posts/conceptos-basicos-sobre-redes-neuronales#:~:text=Epoch,pasaran%20por%20la%20red%20neuronal>.
- <https://enmilocalfunciona.io/deep-learning-basico-con-keras-parte-1/>
- <https://www.ibm.com/docs/es/spss-modeler/SaaS?topic=networks-basics-neural>