Análisis Red Neuronal

Integrantes del grupo:

Jefferson Gutiérrez

Juan Camilo Gallardo

Valeria Larrea

Nicolás Chavez

Andrés Casallas

Red Neuronal

Son un conjunto de algoritmos, que mediante neuronas *artificiales* prentenden imitar el cerebro biológico. Recopilan información en filas llamadas *capas*, las cuales forman conexiones entre si. Su objetivo es que a traves de la experiencia que adquiera asi mismo vaya reconociendo y aprendiendo cosas nuevas, para que en un futuro la información almacenada en las neuronas sea util y de esta forma la **Red Neuronal** sepa llegar a la solución del problema.

Funcionamiento

Su funcionamiento se basa en recibir uno o varios valores de entrada, los cuales seran conectados a un nodo *(Neurona)*, los nodos al tener un valor numérico lograrán modificar los datos de entrada para despues ser validados y poder empezar el corrido entre capas hasta llegar a un valor de salida (*respuesta*).

Utilidad

Hoy en día la utilidad de las **redes neuronales** es inimaginable, puede integrarse desde lo laboral hasta en los temas de ocio. Un buen ejemplo de **red neuronal** es el algoritmo de YouTube, el cual dependiendo de las preferencias del usuario empieza a aprender de ello para posteriormente recomendarle contenido que cumple con sus gustos.

Primera Fórmula:

En esta primera formula cada capa está formada por neuronas. Cada neurona presenta dos conjuntos de parámetros, pesos y sesgos. Cuando los datos de entrada llegan a una neurona, se calcula una función lineal usando los pesos y los sesgos como parámetros.

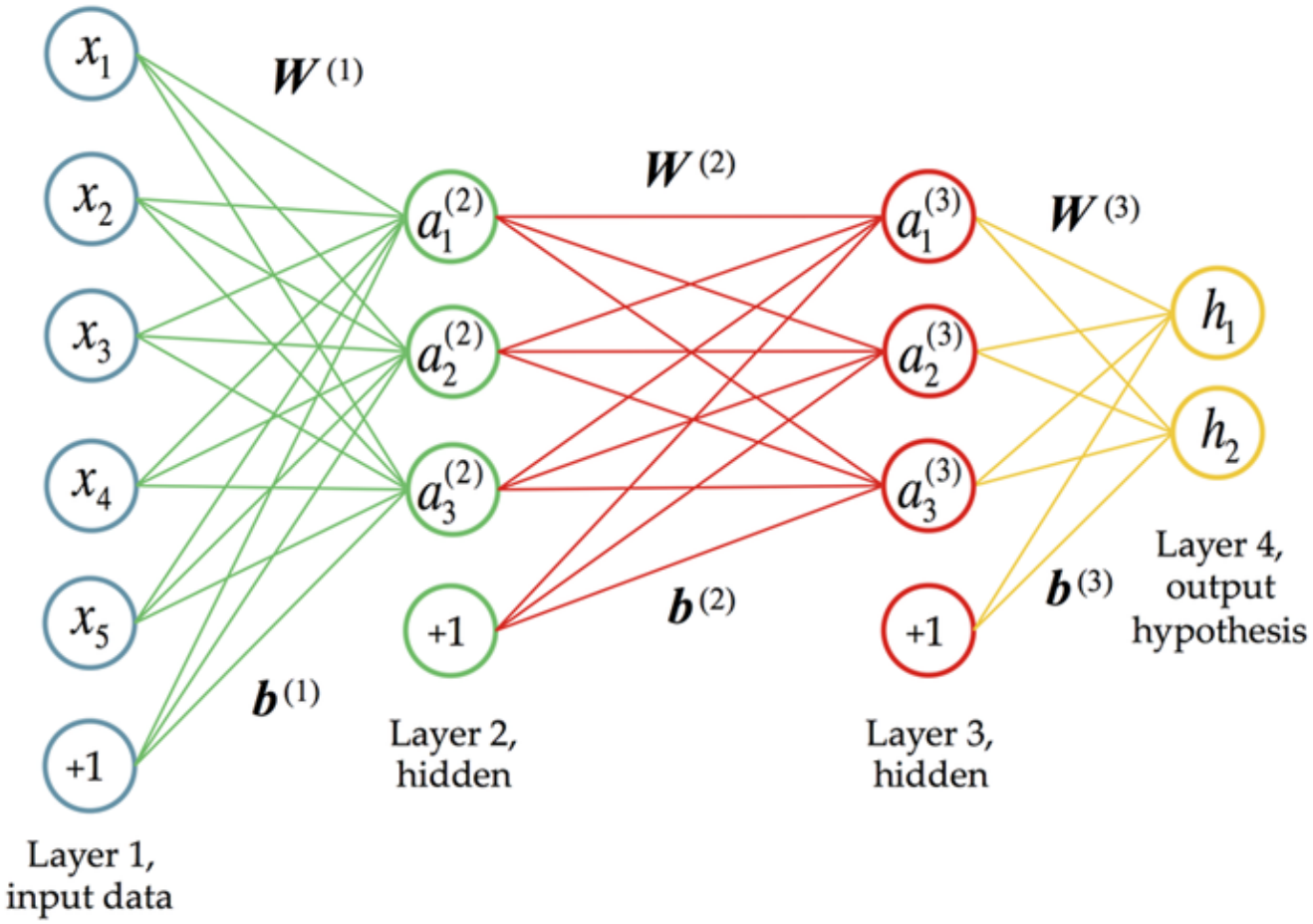
$$Z = W * X + b$$

Es decir: Dónde X, W y b son los datos de entrada, pesos y sesgos respectivamente. Si una **Red neuronal** solamente realizase esta operación, sería análoga a una regresión lineal. Sin embargo, queremos que nuestra red aprenda patrones no lineales, por lo que es necesario aplicar una nueva función a Z.

Segunda Fórmula:

Esta segunda fórmula se conoce como función de activación. Este proceso se repite en cada neurona a través de la red, hasta llegar a la capa de salida.

$$A = g(Z) = g(W * X + b)$$



Perceptrón Multicapa (MLP)

El perceptrón multicapa cuenta con tres tipos de capas principales capa de entrada, capas ocultas y capa de salida, entre las capas ocultas esta la capa que contiene la función de activación.

Capa de entrada: Recibe los datos de entrada. Capa oculta: Cuenta con uno o más neuronas. Capa de activación: Aplica una función de activación sobre la salida de cada neurona de la capa oculta. Capa de salida: Produce la predicción para completar la tarea supervisada. Puede ser de clasificación o regresión.

El MLP usa la cantidad de capas que sean necesarias para su buen funcionamiento.

Tercera Fórmula:

MSE significa Error Cuadrático Medio indicado por sus siglas en inglés. (Medium Cuadratic Error).

Para comparar la salida de la **Red Neuronal** con los datos que queremos predecir y, es necesario introducir una función de pérdida. Un ejemplo de función de pérdida es el error cuadrático medio:

$$MSE = \frac{\sum_{i=1}^{N} (y_i - \widehat{y_i})^2}{N}$$

Dónde $\widehat{y}$ es la salida de la **Red Neuronal**. La función de pérdida es una medida de cuánto difiere nuestra predicción de la realidad. El objetivo de la **Red neuronal** es aproximar las predicciones a los datos y al máximo mediante la minimización de la función de pérdida y el aprendizaje de los parámetros óptimos.

En matemáticas, minimizar una función significa calcular derivadas de la función. Si la función es una función compuesta, es posible utilizar la regla de la cadena para calcular las derivadas con respecto a los distintos parámetros.

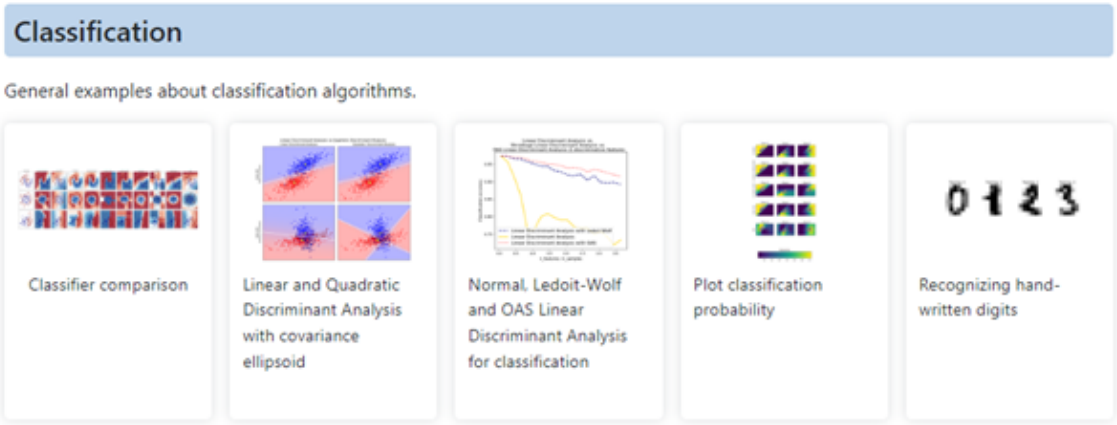
Análisis Scikit-Learn y librerias a utilizar

Primero, debemos comentar que es **Scikit-Learn.**

Scikit-Learn es una librería gratuita hecha en el lenguaje de programación Python la cual incluye técnicas de regresión, clasificación y análisis de grupos. Se puede usar junto con otras librerías como las que veremos a continuación.

Scikit-Learn

Scikit-Learn Una de las funciones protagonistas de **Scikit-Learn** es la clasificación, que hace uso de **Redes Neuronales** en la siguiente imagen podemos ver modelos principales en los que se usa la clasificación.



NumPy

NumPy Es una librería de Python especializada en el cálculo numérico y el análisis de datos, especialmente para un gran volumen de datos. (Incorpora los *Arrays*, los cuales tienen un procesamiento de hasta 50 veces mas rapido que las listas predefinidas de Python)

```
#NumPy

import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)

print(type(arr))
```

```
[1 2 3 4 5]
<class 'numpy.ndarray'>
```

Pandas

Pandas Es una librería de Python especializada en el manejo y análisis de estructuras de datos.

```
#Pandas

import pandas as pd
s = pd.Series({'Matemáticas': 6.0,  'Economía': 4.5, 'Programación': 8.5})
print(s)
```

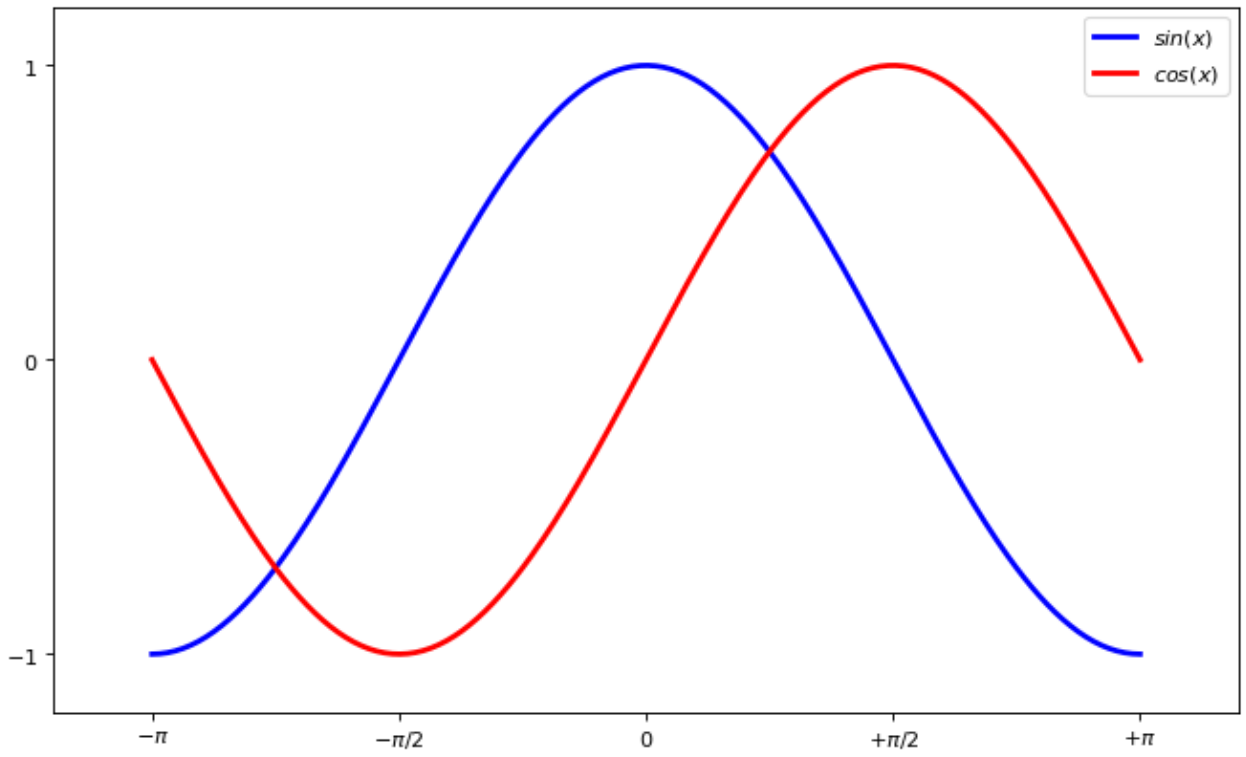
```
Matemáticas     6.0
Economía        4.5
Programación    8.5
dtype: float64
```

Matplotlib

Matplotlib Es una librería de Python especializada en la creación de gráficos en dos dimensiones. Permite crear y personalizar los tipos de gráficos más comunes.

```
#Matplotlib

import numpy as np
from matplotlib import pyplot as plt

plt.figure(figsize=(10,6), dpi=80)
```

```
x = np.linspace(-np.pi, np.pi, 256,endpoint=True)
C,S = np.cos(x), np.sin(x)

plt.plot(x, C, color="blue", linewidth=2.5, linestyle="-", label=r'$sin(x)$')
plt.plot(x, S, color="red",  linewidth=2.5, linestyle="-", label=r'$cos(x)$')
plt.xlim(x.min()*1.2, x.max()*1.2)
plt.ylim(C.min()*1.2, C.max()*1.2)
plt.xticks([-np.pi, -np.pi/2, 0, np.pi/2, np.pi],
          [r'$-\pi$', r'$-\pi/2$', r'$0$', r'$+\pi/2$', r'$+\pi$'])
plt.yticks([-1,0,1],
          [r'$-1$', r'$0$', r'$1$'])

plt.legend()
plt.show()
```



Dato Curioso

Estas librerias, en especial **Scikit-Learn**, son usadas hoy en día por grandes compañias en sus proyectos y aplicaciones para solventar sus necesidades y las de sus usuarios, un par de ejemplos a tener en cuenta:

JP Morgan

"Scikit-learn es una parte indispensable del conjunto de herramientas de aprendizaje automático de Python en JPMorgan. Se usa ampliamente en todas las partes del banco para clasificación, análisis predictivo y muchas otras tareas de aprendizaje automático. Su API sencilla, su amplitud de algoritmos y la calidad de su documentación se combinan para hacer que scikit-learn sea simultáneamente muy accesible y muy poderoso."

Stephen Simmons, VP, Athena Research, JPMorgan

Spotify

"Scikit-learn proporciona una caja de herramientas con implementaciones sólidas de un montón de modelos de última generación y facilita su integración en las aplicaciones existentes. Lo hemos estado usando bastante para recomendaciones musicales en Spotify y creo que es el paquete ML mejor diseñado que he visto hasta ahora."

Erik Bernhardsson, Engineering Manager Music Discovery & Machine Learning, Spotify

Muchas Gracias Fin del proyecto.