Taller de Inteligencia Artificial con Python

Dr. León Felipe Dozal García

Objetivo

En este talle de IA con Python, explicaremos algunos conceptos fundamentales de la inteligencia artificial (IA) y practicaremos con el lenguaje de programación Python. Este taller está diseñado para satisfacer sus necesidades de aprendizaje y ofrece un enfoque paso a paso para usar algunas técnicas de IA con Python.

Desde comprender conceptos básicos hasta explorar aplicaciones y algoritmos avanzados, este taller le proporcionará las habilidades y conocimientos esenciales para crear una base sólida en el uso de IA con Python.

¿Por qué usar Python para IA?

Python proporciona una sintaxis clara y legible, por lo que proporciona un camino sencillo para aprender y crear modelos inteligentes sin estructuras de código complejas. La mejor parte de usar Python es su rico ecosistema de bibliotecas y marcos especialmente diseñados para IA y aprendizaje automático. Python tiene una sólida comunidad de entusiastas, investigadores y desarrolladores de IA que comparten conocimientos, ideas y recursos. El espíritu colaborativo de la comunidad Python AI garantiza que la ayuda esté siempre a su alcance.

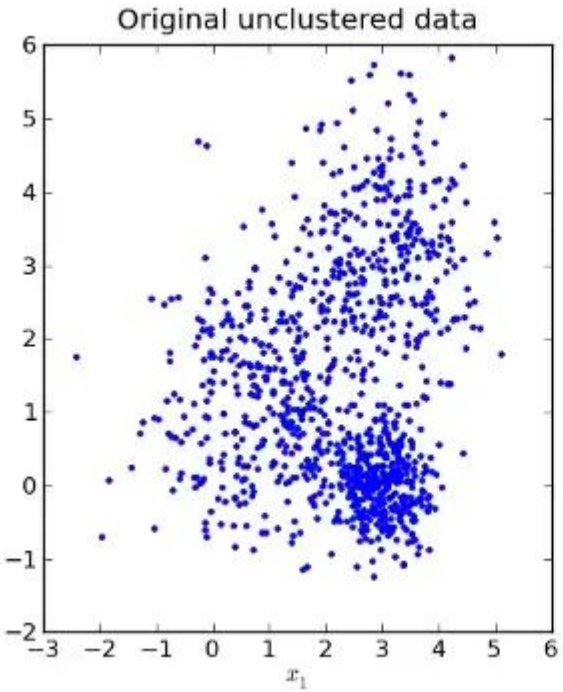
Contenido y Duración

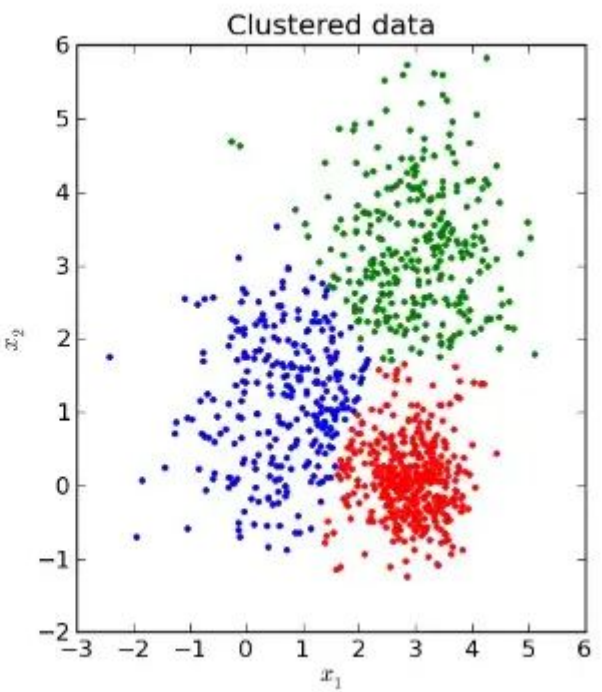
1. Agrupación (Clustering)
2. Aprendizaje profundo y visión por computadora
3. Algoritmos genéticos

Agrupación (clustering)

Agrupación (clustering)

La agrupación es un método de aprendizaje automático NO SUPERVISADO y consiste en agrupar datos SIN ETIQUETAR en función de sus similitudes. El objetivo de la agrupación es identificar patrones y relaciones en los datos sin ningún conocimiento previo del significado de los datos.





Agrupación (clustering)

Existen diferentes tipos de algoritmos de agrupamiento en clústeres.

- K-medias (K-means): Particiona los datos en K grupos (clusters).
- Jerárquica (Hierarchical): Construye una estructura jerárquica de grupos.
- Basada en Densidad (Density-Based, DBSCAN): Identifica grupos basado en la densidad de los datos.
- Cambio-medio (Mean-Shift): Encuentra grupos basado en procesos de búsqueda.
- Espectral (Spectral): Usa la teoría espectral de grafos para agrupar datos.

Segmentación de clientes usando Agrupación Jerárquica.

¿Qué es la segmentación de clientes?

“La segmentación de los clientes consiste en identificar y organizar grupos de compradores usando variables y características específicas que tienen en común. Estos criterios de segmentación de los clientes pueden ser rasgos de la personalidad, demografía, geografía, o incluso sus ingresos.”

https://www.freshworks.com/latam/crm/segmentacion-de-clientes/

¿Para qué sirve la segmentación de clientes?

“La segmentación ofrece información detallada de los consumidores que ayuda a los especialistas de marketing a adaptar sus productos y servicios a las necesidades de estos. Esta personalización proporciona una ventaja competitiva y una mayor posibilidad de lograr la conversión de clientes y lealtad de marca.”

https://www.freshworks.com/latam/crm/segmentacion-de-clientes/

¿Cómo hacer una segmentación de clientes?

“Para sacar el máximo provecho de las campañas de marketing, es fundamental realizar una investigación exhaustiva de segmentación de clientes. Para subdividir de forma efectiva el mercado objetivo en segmentos, es necesario recopilar datos de consumidores, clasificarlos sobre la base de variables estándar, analizar el potencial de cada segmento, desarrollar estrategias para ingresar en ellos, personalizar la oferta y, finalmente, medir el desempeño de las campañas.”

https://www.freshworks.com/latam/crm/segmentacion-de-clientes/

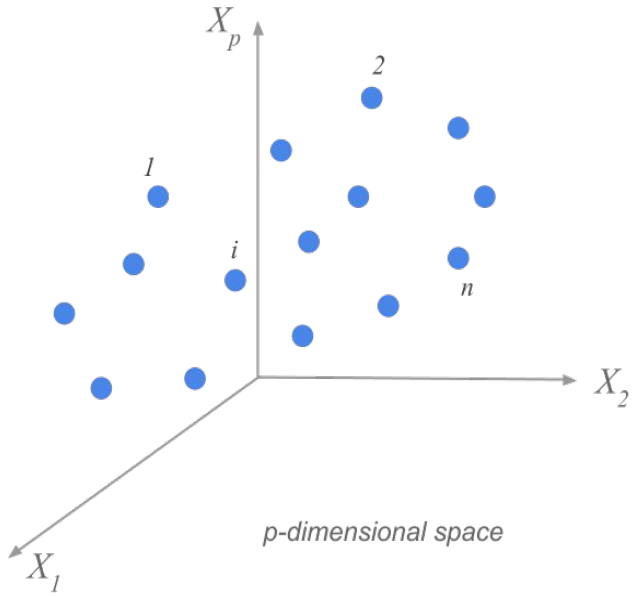
Problema

El gerente de un centro comercial desea hacer una segmentación para comprender a los clientes y poder darle información al equipo de marketing y planificar la estrategia en consecuencia.

Agrupación Jerárquica

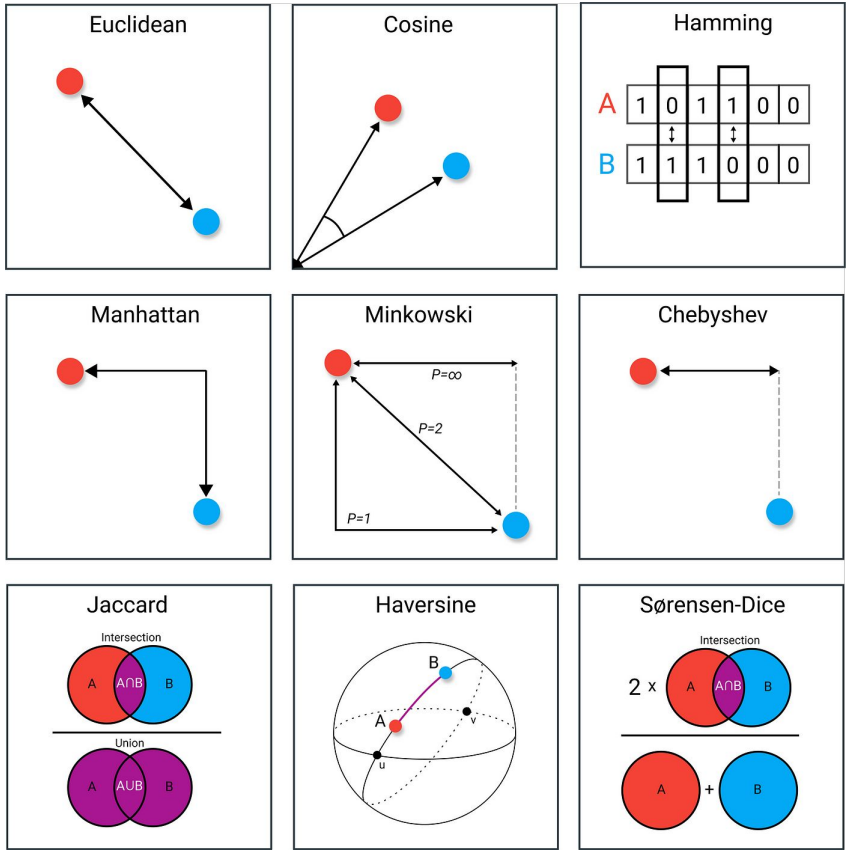
Dado un conjunto de datos de clientes conformados por los valores X_1, X_2 y X_p.

	X_1	X_2	X_p
1			
2			
⋮			
i			
⋮			
n			



Medidas de similitud o distancia

La agrupación jerárquica es un método de agrupación basado en una medida arbitraria de similitud o distancia entre cualquier par de puntos de datos. De manera que entre más cerca estén dos puntos de datos mayor será su similitud, o viceversa. Este algoritmo de agrupación no requiere que especifiquemos previamente el número de agrupaciones.



Medidas de similitud y distancia

Camberra :

$$d(x,y) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|} \quad (2)$$

Minkowsky :

$$d(x,y) = \left(\sum_{i=1}^{m} |x_i - y_i|^r \right)^{1/r} \quad (3)$$

Chebychev :

$$d(x,y) = \max_{i=1}^{m} |x_i - y_i|$$

Euclidean :

$$d(x,y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2} \quad (5)$$

Manhattan / city - block :

$$d(x,y) = \sum_{i=1}^{m} |x_i - y_i| \quad (6)$$

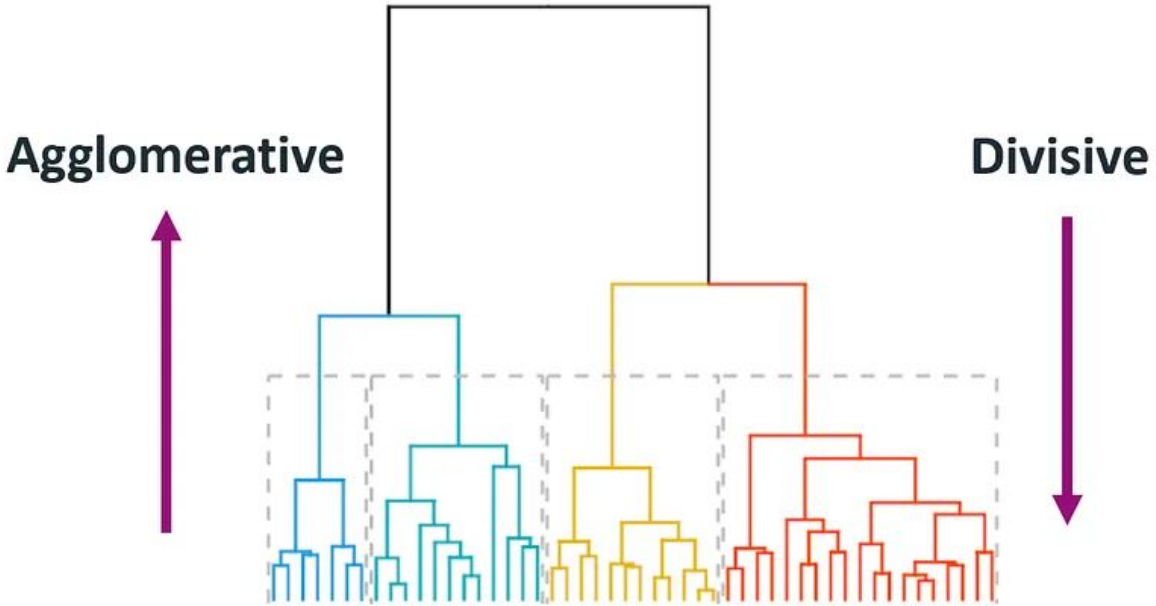
Measure	Formula	Measure	Formula
Cosine Similarity	$\frac{\sum x_i y_i}{\sqrt{\sum x_i^2}\sqrt{\sum y_i^2}}$	Jaccard's Index	$\frac{\lvert X \cap Y \rvert}{\lvert X \cup Y \rvert}$
Manhattan Distance	$\sum \lvert x_i - y_i \rvert$	Dice's Coefficient	$2\frac{\lvert X \cap Y \rvert}{\lvert X \rvert + \lvert Y \rvert}$
Euclidean Distance	$\sqrt{\sum \lvert x_i - y_i \rvert^2}$		

Tipos de Agrupamiento Jerárquico

1. Aglomerativos (ascendentes). Al comienzo, estos algoritmos tratan cada dato como un clúster único y luego aglomeran sucesivamente los pares de clusters más similares hasta que todos los clusters se han fusionado en un solo clúster que contiene todos los datos.
2. Divisivos (descendentes). Al comienzo, un clúster contiene todos los datos y divide los clusters de forma recursiva hasta que los datos individuales se hayan dividido en clusters únicos.

Tipos de Agrupamiento Jerárquico

Un dendrograma, una figura parecida a un árbol producida por agrupamiento jerárquico, representa las relaciones jerárquicas entre grupos. Para generar diferentes números de grupos, el dendrograma se puede cortar a varias alturas.



Pseudocódigo del agrupamiento jerárquico aglomerativo

Entrada: D (conjunto de datos individuales)
Salida: C (conjunto de datos divididos en clusters)

Inicializar $\ell = 0$
Inicializar el conjunto de clusters $C_t = D$
mientras $|C_t| > 1$ **hacer**
Encontrar $c_i, c_j \in C_t$ tal que la distancia $d(c_i, c_j)$ sea mínima.
Crear $C_* = c_i \cup c_j$ como padre de los clusters c_i y c_j
Actualizar $C_{t+1} = C_t - \{c_i, c_j\} \cup c_*$
$\ell = \ell + 1$
fin mientras
regresar C

Base de datos de clientes

Datos de clientes de un centro comercial.

Este archivo de entrada contiene la información básica (identificación, edad, sexo, ingresos y puntuación de gastos) sobre los clientes de un centro comercial. El puntaje de gasto es algo que usted asigna al cliente en función de sus parámetros definidos, como el comportamiento del cliente y los datos de compra.

Abrir el archivo:

… /TallerIA/Hierarchical Clustering/Datos/Mall_Customers.csv

CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40

Visualización de Datos

Visualización de datos

Abrir, modificar ruta de datos y ejecutar el archivo:

… /Taller_IA/Hierarchical Clustering/Visualizacion_Datos.py

Usaremos la siguiente paqueteria:

- Pandas
- Matplotlib
- Seaborn
- Sklearn

Pandas

Pandas es una biblioteca del lenguaje de programación Python, enteramente dedicada a la ciencia de datos. El nombre “Pandas” es en realidad una contracción del término “Panel Data” para conjuntos de datos. Esta biblioteca fue creada como una herramienta de alto nivel para cargar, alinear, manipular o fusionar datos estructurados y no estructurados. Es potente, flexible y fácil de usar.

Los creadores de Pandas planean hacer evolucionar esta biblioteca para convertirla en la herramienta de manipulación y análisis de datos de código abierto más potente y flexible en cualquier lenguaje de programación.

https://pandas.pydata.org/

Matplotlib

Matplotlib es una biblioteca de trazado de gráficos de bajo nivel en Python que sirve para la visualización de datos. Fue creada por John D. Hunter y su código es abierto por lo que podemos usarla libremente.

Matplotlib está escrita principalmente en Python, algunos segmentos están escritos en C, Objective-C y Javascript para compatibilidad con la plataforma.

El código fuente de Matplotlib se encuentra en este repositorio de github https://github.com/matplotlib/matplotlib

Seaborn

Es una biblioteca para crear gráficos estadísticos en Python. Está basado en Matplotlib y se integra con las estructuras de Pandas.

Esta biblioteca es tan poderosa como Matplotlib pero aporta simplicidad y nuevas funciones. Nos permite explorar y comprender datos rápidamente.

Las funciones internas para mapeo semántico y agregación estadística permiten convertir los datos en visualizaciones gráficas.

Seaborn abstrae toda la complejidad de Matplotlib. Sin embargo, es posible crear gráficos que satisfagan todas sus necesidades y requisitos.

https://datascientest.com/en/seaborn_and_data_visualization

Sklearn

Es una biblioteca de aprendizaje máquina. Contiene herramientas simples y eficientes para el análisis predictivo de datos. Es fácil de usar, construido sobre NumPy, SciPy, and Matplotlib. Es Open Source.

Entre las herramientas que ofrecen están varios algoritmos de clasificación, regresión numérica, agrupamiento, reducción de dimensionalidad, selección de modelos y preprocesamiento de datos.

https://scikit-learn.org/stable/

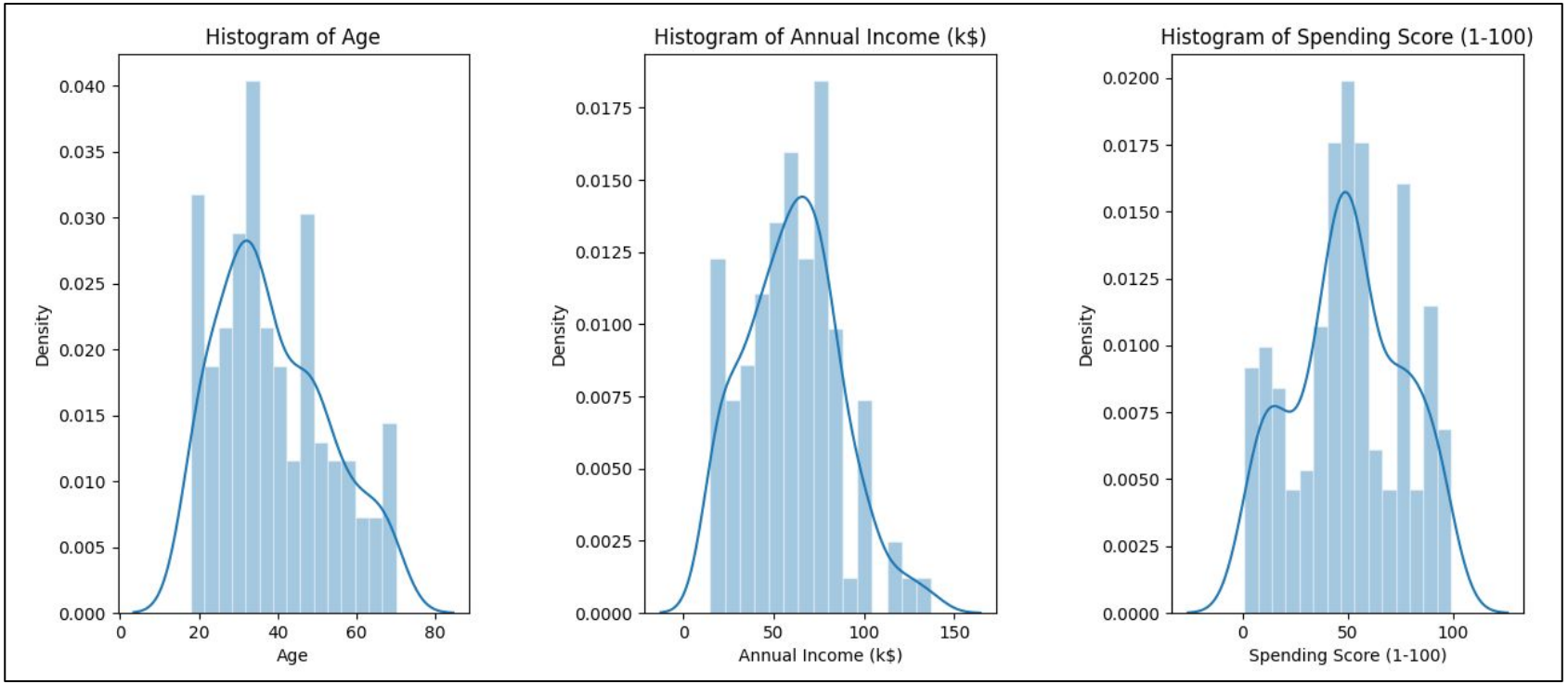
Visualización de datos y estadísticas

CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
1	Male	19	15	39
2	Male	21	15	81
3	Female	20	16	6
4	Female	23	16	77
5	Female	31	17	40

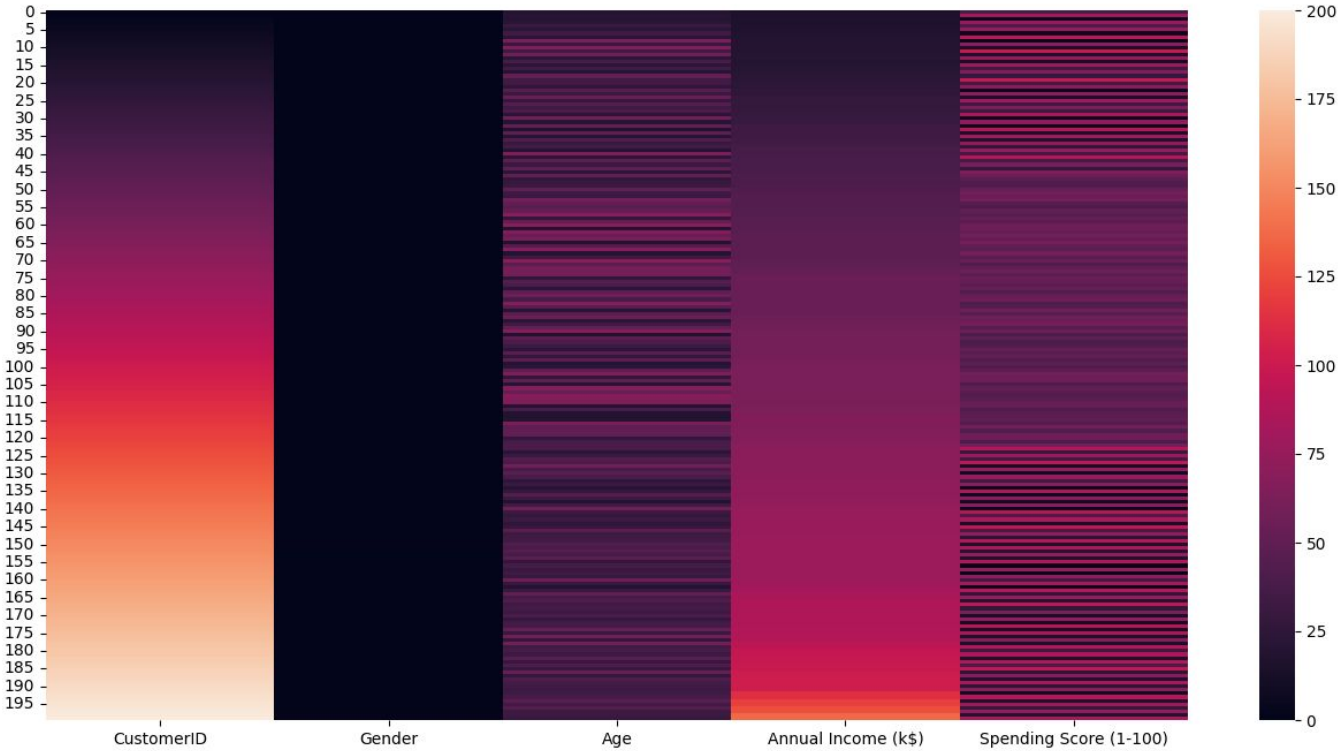
```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

	CustomerID	Age	Annual Income (k$)	Spending Score (1-100)
count	200.000000	200.000000	200.000000	200.000000
mean	100.500000	38.850000	60.560000	50.200000
std	57.879185	13.969007	26.264721	25.823522
min	1.000000	18.000000	15.000000	1.000000
25%	50.750000	28.750000	41.500000	34.750000
50%	100.500000	36.000000	61.500000	50.000000
75%	150.250000	49.000000	78.000000	73.000000
max	200.000000	70.000000	137.000000	99.000000

Histograma y diagrama de densidad



Mapa de calor



Agrupamiento Jerárquico Aglomerativo

Agrupamiento Jerárquico Aglomerativo

Abrir, modificar ruta de datos y ejecutar el archivo:

… /Taller_IA/Hierarchical Clustering/Agrupamiento_Jerarquico.py

Usaremos la siguiente paqueteria:

- Pandas
- Matplotlib
- Seaborn
- Sklearn
- Ploty
- SciPy

Ploty

La biblioteca de gráficos Python de Plotly crea gráficos interactivos con calidad de publicación. Ejemplos de cómo crear diagramas de líneas, diagramas de dispersión, gráficos de áreas, gráficos de barras, barras de error, diagramas de cajas, histogramas, mapas de calor, subtramas, gráficos de ejes múltiples, gráficos polares y gráficos de burbujas.

Plotly.py es gratuito y de código abierto y puede ver el código fuente, informar problemas o contribuir en GitHub.

https://plotly.com/python/

SciPy

SciPy proporciona algoritmos para optimización, integración, interpolación, problemas de valores propios, ecuaciones algebraicas, ecuaciones diferenciales, estadísticas y muchas otras clases de problemas. Extiende a NumPy proporcionando herramientas adicionales para la computación de matrices y proporciona estructuras de datos especializadas, como matrices dispersas y árboles k-dimensionales.

SciPy incluye implementaciones altamente optimizadas escritas en lenguajes de bajo nivel como Fortran, C y C++.

La sintaxis de alto nivel de SciPy lo hace accesible y productivo para programadores de cualquier nivel o experiencia. Además, es Open Source.

https://scipy.org/

Dendrograma con distancia euclidiana

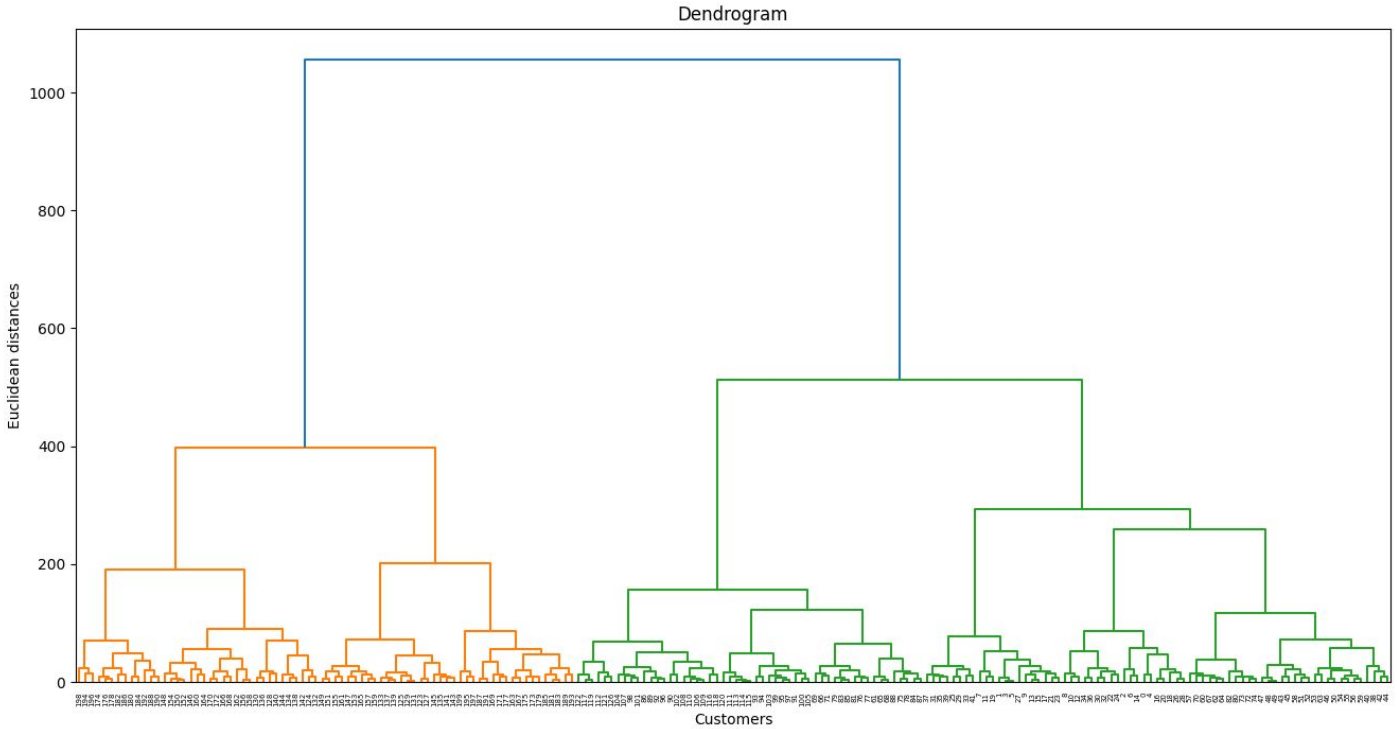


Diagrama de Clusters con 3 variables

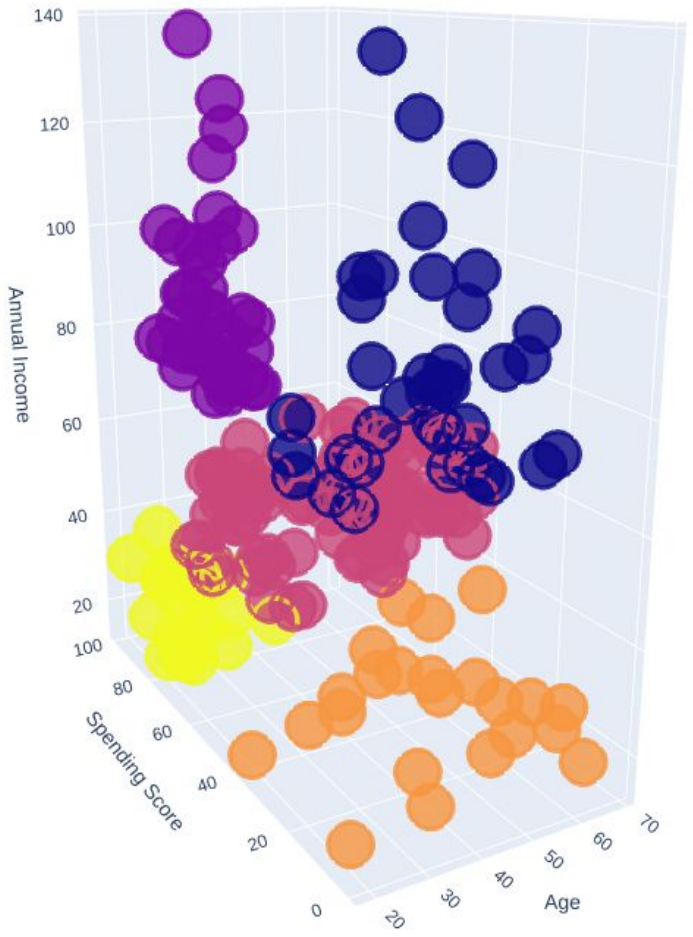
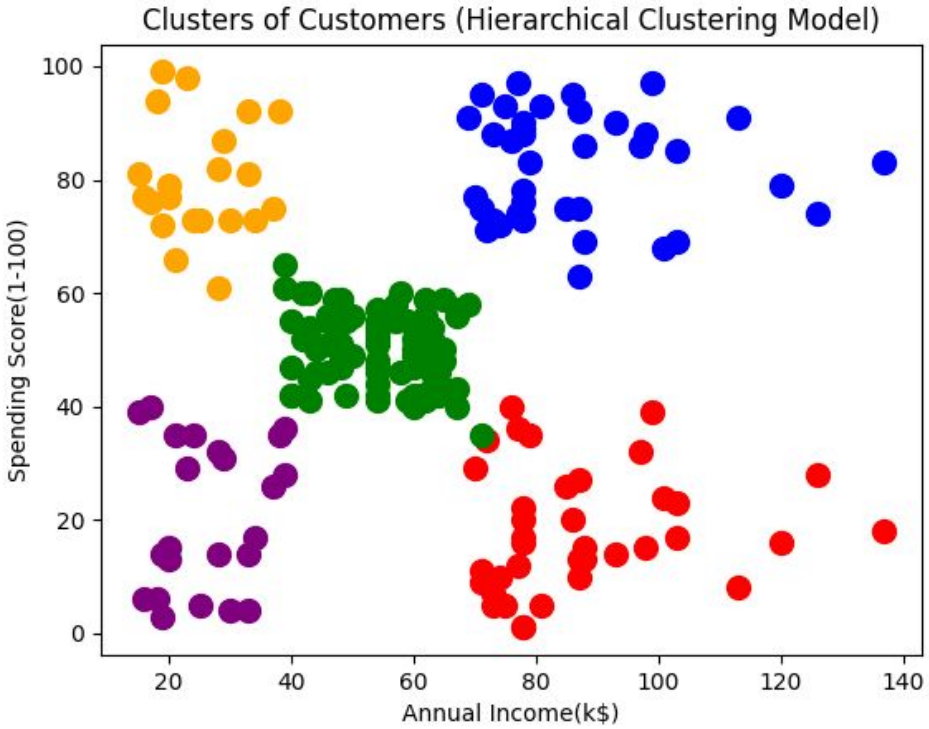


Diagrama de Clusters con 2 variables



Resultados

Abrir el archivo de resultados:

… /Taller_IA/Hierarchical Clustering/Datos/segmented_customers.csv

Referencias

Clustering

https://www.geeksforgeeks.org/ml-types-learning-part-2/

Hierarchical clustering

https://www.geeksforgeeks.org/hierarchical-clustering/

Práctica

https://www.kaggle.com/code/heeraldedhia/hierarchical-clustering-for-customer-data

Base de datos

https://github.com/SteffiPeTaffy/machineLearningAZ/blob/master/Machine%20Learning%20A-Z%20Template%20Folder/Part%204%20-%20Clustering/Section%2025%20-%20Hierarchical%20Clustering/Mall_Customers.csv

Aprendizaje Profundo y Visión por Computadora

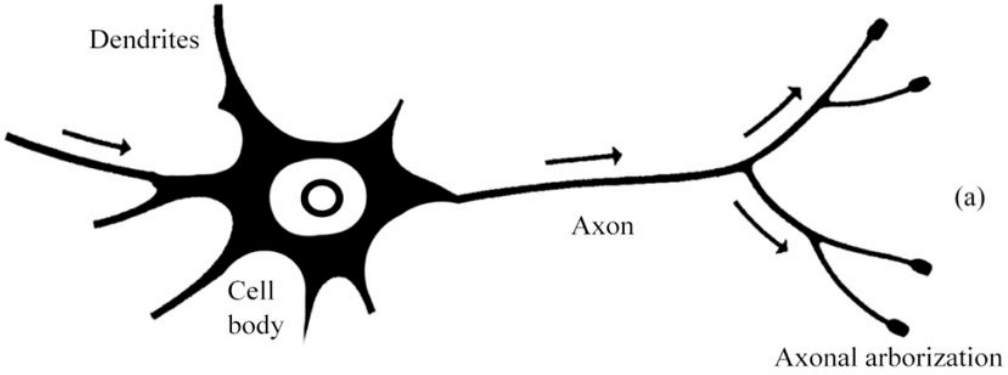
Aprendizaje Profundo

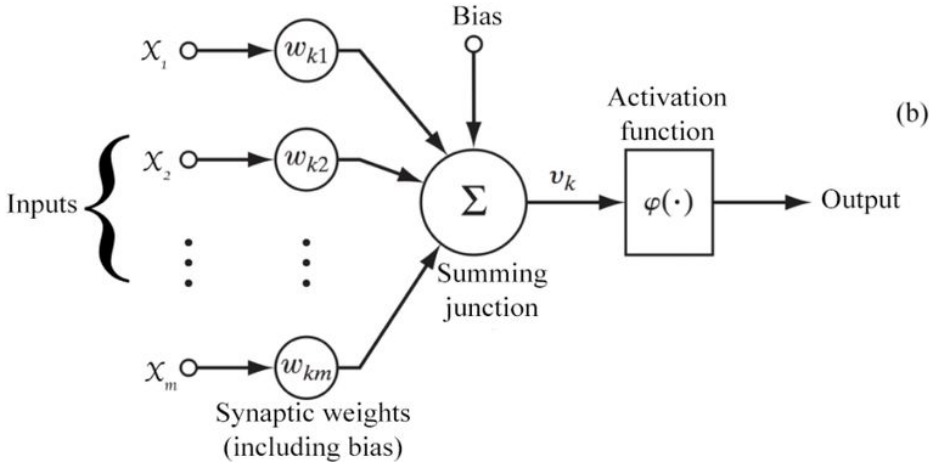
El aprendizaje profundo es un subcampo del aprendizaje automático. El modelo de aprendizaje profundo se inspira en la estructura del cerebro humano. El cerebro humano está formado por miles de millones de neuronas que se comunican a través de señales electroquímicas y en el Aprendizaje Profundo, las redes neuronales artificiales están compuestas por nodos que están interconectados a través de conexiones ponderadas.

El término "profundo" se refiere a las múltiples capas (profundidad) de estas redes, lo que les permite aprender patrones y características intrincados a partir de vastos conjuntos de datos.

Fundamentos de Aprendizaje Profundo

Perceptrón

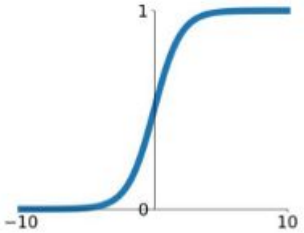

Dendrites
Axon
Cell
body
Axonal arborization
(a)


Bias
x_1
w_{k1}
x_2
w_{k2}
Inputs
x_m
w_{km}
Σ
v_k
Activation
function
$\varphi(\cdot)$
Output
Summing
junction
Synaptic weights
(including bias)
(b)

Funciones de activación

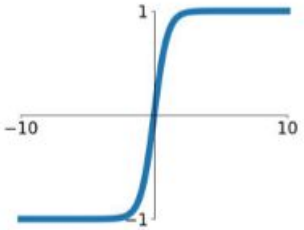
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



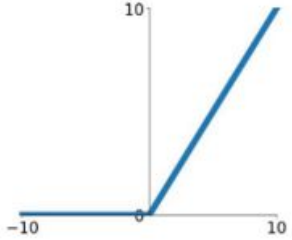
tanh

$$\tanh(x)$$



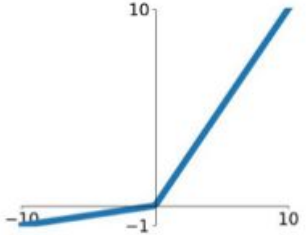
ReLU

$$\max(0, x)$$



Leaky ReLU

$$\max(0.1x, x)$$

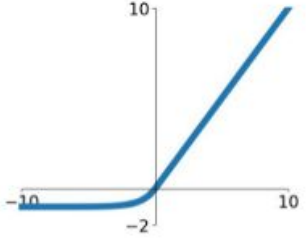


Maxout

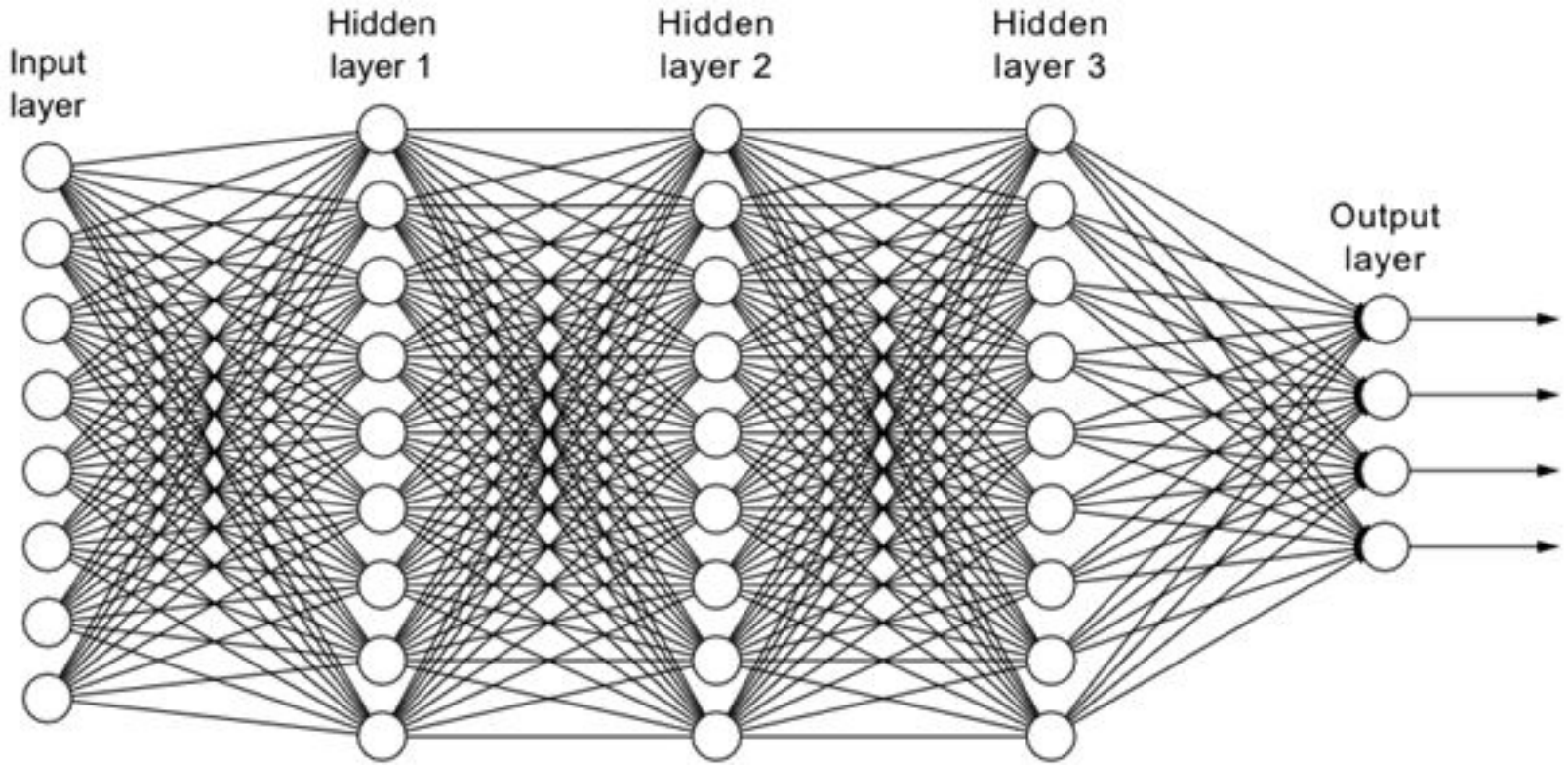
$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



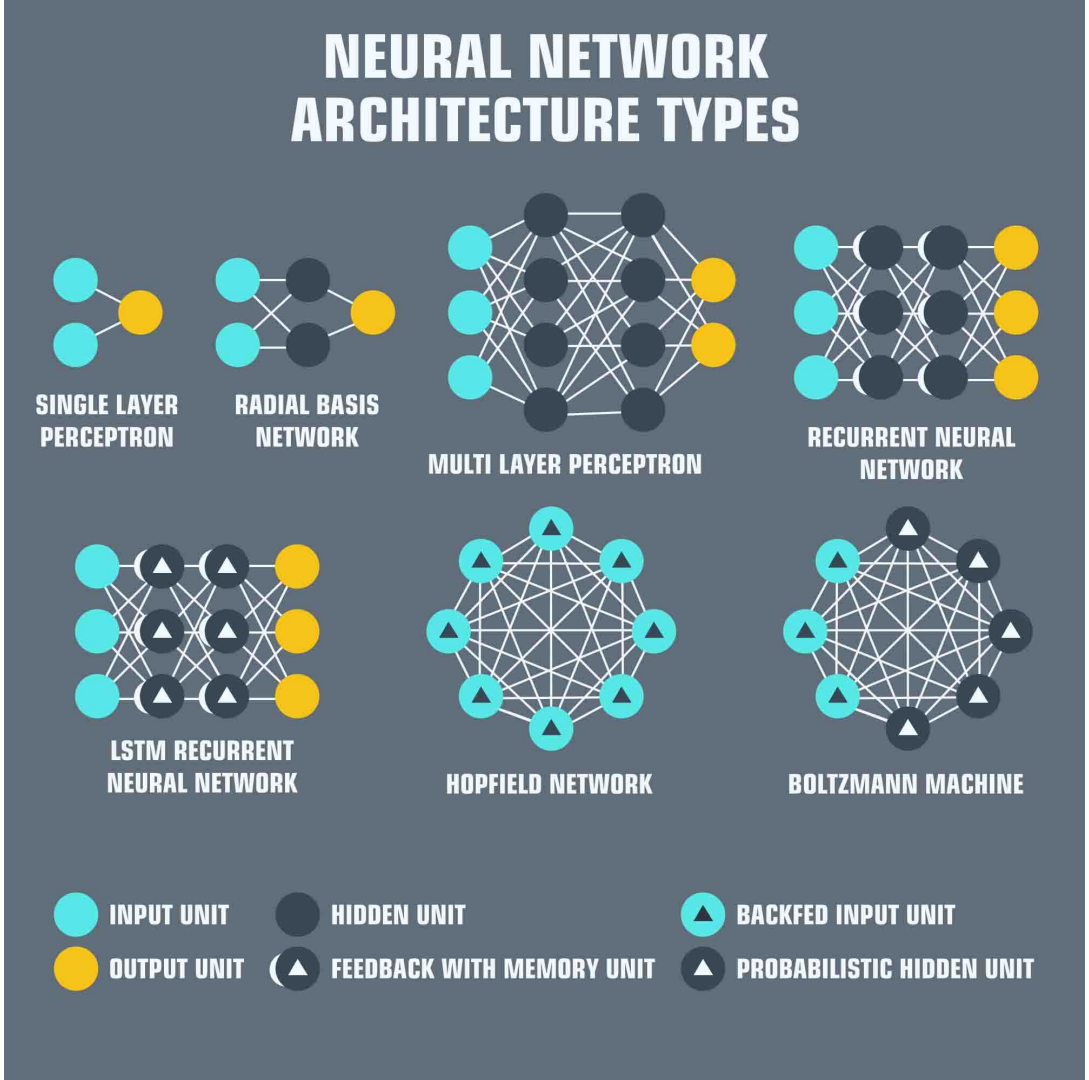
Redes Neuronales Artificiales



Arquitectura de aprendizaje

Las arquitecturas de aprendizaje profundo son modelos de redes neuronales diseñados para facilitar tareas de aprendizaje complejas mediante la identificación automática de patrones y representaciones dentro de los datos.

A continuación se muestran algunas estructuras de aprendizaje profundo:



Algoritmos de entrenamiento

Piense en cómo una máquina aprende a partir de los datos durante el entrenamiento. Se le llama entrenamiento al proceso que determina el valor de los pesos y bias asociados a cada conexión (parámetros), a partir de muchos datos etiquetados, con el fin de que la red neuronal realice alguna tarea determinada.

- Gradient Descent
- Backpropagation
- Adam

Hiperparámetros

Los hiperparámetros son parámetros cuyos valores controlan el proceso de aprendizaje y determinan los valores de los parámetros del modelo que un algoritmo de aprendizaje termina aprendiendo. El prefijo 'hyper' sugiere que son parámetros de 'nivel superior' que controlan el proceso de aprendizaje y los parámetros del modelo que resultan de él.

- Tamaño de los conjuntos de entrenamiento y validación.
- Algoritmo de entrenamiento (e.g., gradient descent, stochastic gradient descent).
- Taza de aprendizaje en el algoritmo de entrenamiento.

Hiperparámetros

- Función de activación en cada capa (e.g. Sigmoid, ReLU, Adam).
- Número de capas ocultas.
- Número de neuronas en cada capa.
- Número de iteraciones (épocas) en el entrenamiento.
- Tamaño del kernel en capas convolucionales.
- Tamaño del Pooling.

Modelo de Clasificación de Imágenes

¿Qué es la clasificación de imágenes?

Es la tarea de asignar una etiqueta, o clase, a una imágen basados en su contenido.

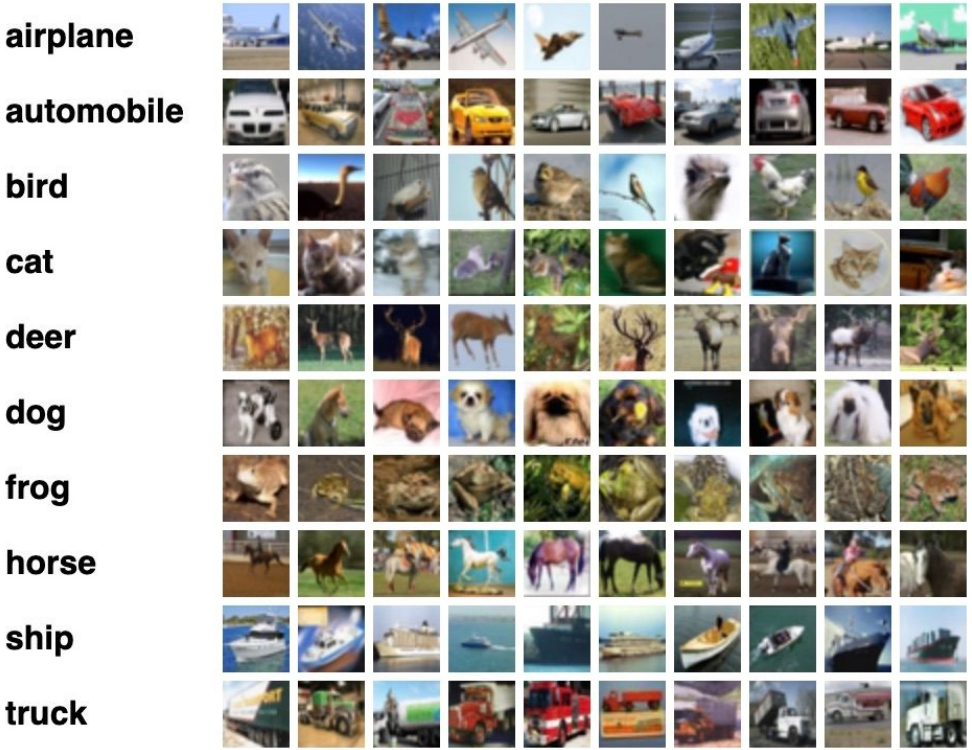
Base de Imágenes

CIFAR-10 de Keras

El conjunto de datos consta de 60000 imágenes en color de 32 x 32 píxeles en 10 clases, con 6000 imágenes por clase. Hay 50000 imágenes de entrenamiento y 10000 imágenes de prueba.

El conjunto de datos se divide en cinco lotes de entrenamiento y un lote de prueba, cada uno con 10000 imágenes. El lote de prueba contiene exactamente 1000 imágenes seleccionadas al azar de cada clase. Los lotes de entrenamiento contienen las imágenes restantes en orden aleatorio, pero algunos lotes de entrenamiento pueden contener más imágenes de una clase que de otra. Entre ellos, los lotes de entrenamiento contienen exactamente 5000 imágenes de cada clase.

CIFAR-10



https://www.cs.toronto.edu/~kriz/cifar.html

Práctica

Colab

1. Abrir Colab
2. Abrir el bloc de notas: … /Taller_IA/Aprendizaje Profundo/Clasificacion_Imagenes.ipynb

```
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras.utils import to_categorical
from keras.preprocessing import image
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from keras.utils import to_categorical

# Montar Google Drive
from google.colab import drive
drive.mount('/content/drive')

(x_train, y_train), (x_test, y_test) = keras.datasets.cifar10.load_data()

x_train.shape,x_test.shape
```

```
Mounted at /content/drive
Downloading data from https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz
170498071/170498071 [==============================] - 2s 0us/step
((50000, 32, 32, 3), (10000, 32, 32, 3))
```

Keras

El propósito de Keras es brindar una ventaja cualquier desarrollador que busque ofrecer aplicaciones basadas en aprendizaje profundo. Keras se centra en la velocidad de depuración, la elegancia y concisión del código, la mantenibilidad y la capacidad de implementación. Con Keras, tu código base es más pequeño, más legible y más fácil de iterar. Sus modelos se ejecutan más rápido gracias a la compilación, y son más fáciles de implementar en todas las superficies (servidor, móvil, navegador, integrados) gracias a los componentes de servicio de los ecosistemas TensorFlow y PyTorch.

https://keras.io/

Numpy

NumPy es una biblioteca de Python que proporciona soporte para arreglos multidimensionales y funciones matemáticas de alto rendimiento. Con numpy, podemos realizar operaciones numéricas de manera eficiente y realizar cálculos complejos de forma sencilla. Utilizar numpy tiene numerosas ventajas. Algunas de ellas son:

- Eficiencia: utiliza arreglos optimizados en memoria para realizar cálculos rápidos.
- Funciones matemáticas avanzadas: proporciona una amplia gama de funciones matemáticas para realizar operaciones complejas.
- Integración con otras bibliotecas: se integra fácilmente con otras bibliotecas de Python, como matplotlib y pandas, para realizar análisis de datos y visualizaciones.

https://numpy.org/

Preprocesamiento de imágenes

```
# Normalización
x_train = x_train/255.0
x_test = x_test/255.0

# Promediar las tres bandas RGB para obtener una imagen es tonos de gris
x_train = np.mean(x_train, axis=3)
x_test = np.mean(x_test, axis=3)
x_train.shape,x_test.shape
```

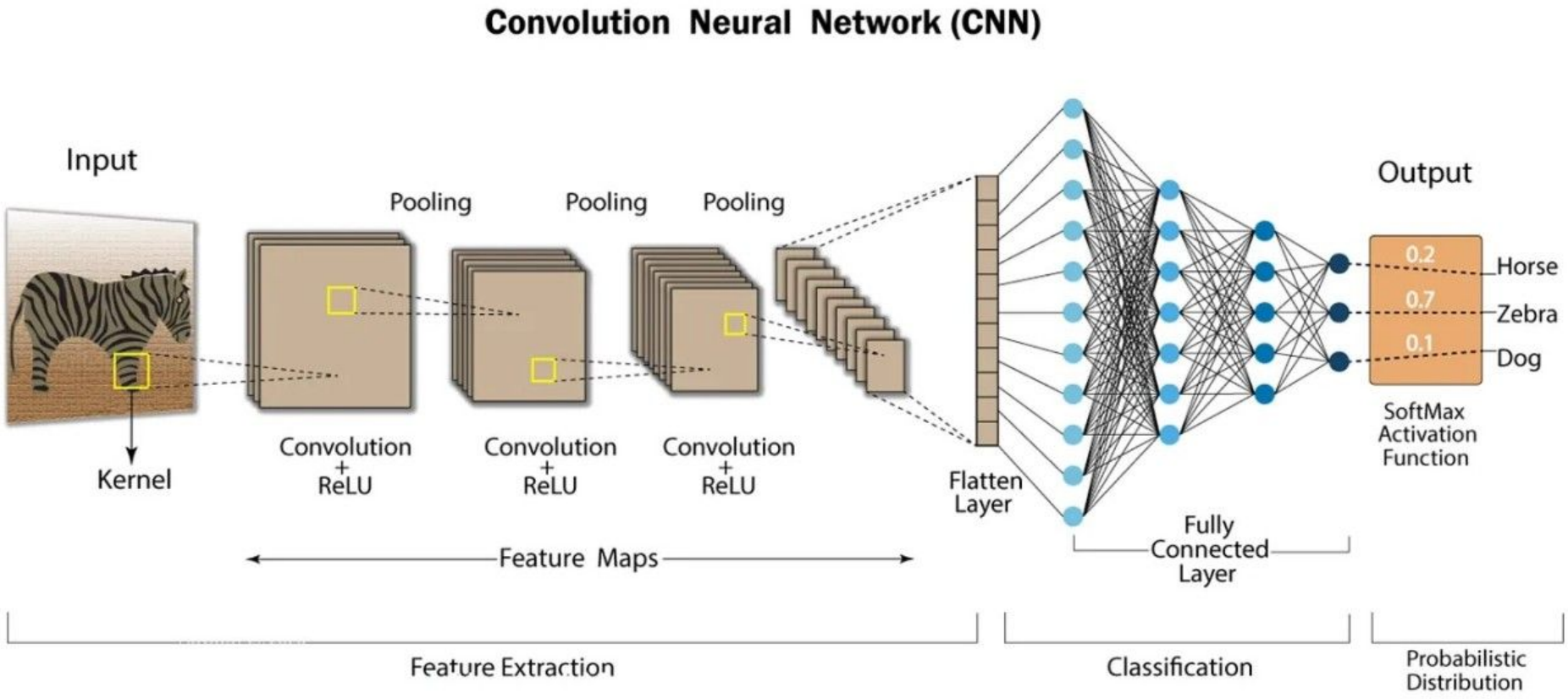
```
((50000, 32, 32), (10000, 32, 32))
```

Modelo de red profunda

```
# Definir la estructura del modelo
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3),activation='relu',input_shape=(32,32,1)))
model.add(Conv2D(64, (3, 3), activation='relu'))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(128, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(10, activation='softmax'))

# Compilar el modelo
model.compile(loss='sparse_categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])
```

Modelo de red profunda



Entrenamiento

```
# Entrenar el modelo
model.fit(x_train, y_train, epochs=10, validation_data=(x_test, y_test))
```

```
Epoch 1/10
1563/1563 [==============================] - 201s 128ms/step - loss: 1.7540 - accuracy: 0.3652 - val_loss: 1.3796 - val_accuracy: 0.5173
Epoch 2/10
1563/1563 [==============================] - 204s 130ms/step - loss: 1.4321 - accuracy: 0.4891 - val_loss: 1.2319 - val_accuracy: 0.5707
Epoch 3/10
1563/1563 [==============================] - 213s 136ms/step - loss: 1.3133 - accuracy: 0.5340 - val_loss: 1.1326 - val_accuracy: 0.6057
Epoch 4/10
1563/1563 [==============================] - 193s 123ms/step - loss: 1.2373 - accuracy: 0.5630 - val_loss: 1.1011 - val_accuracy: 0.6180
Epoch 5/10
1563/1563 [==============================] - 191s 123ms/step - loss: 1.1725 - accuracy: 0.5900 - val_loss: 1.0570 - val_accuracy: 0.6390
Epoch 6/10
1563/1563 [==============================] - 190s 122ms/step - loss: 1.1206 - accuracy: 0.6076 - val_loss: 1.0355 - val_accuracy: 0.6402
Epoch 7/10
1563/1563 [==============================] - 188s 120ms/step - loss: 1.0808 - accuracy: 0.6182 - val_loss: 1.0247 - val_accuracy: 0.6454
Epoch 8/10
1563/1563 [==============================] - 192s 123ms/step - loss: 1.0345 - accuracy: 0.6352 - val_loss: 1.0041 - val_accuracy: 0.6486
Epoch 9/10
1563/1563 [==============================] - 200s 128ms/step - loss: 1.0083 - accuracy: 0.6443 - val_loss: 1.0111 - val_accuracy: 0.6543
Epoch 10/10
1563/1563 [==============================] - 194s 124ms/step - loss: 0.9661 - accuracy: 0.6567 - val_loss: 0.9733 - val_accuracy: 0.6630
<keras.src.callbacks.History at 0x7cac56e1ee90>
```

Predicciones

Preprocesamiento de imágenes

1. Abrir el archivo … /Taller_IA/Aprendizaje Profundo/Preprocesar_imagen.py
2. Actualizar las ubicaciones de archivos en el código.
3. Descargar imágenes representativas de alguna de las 10 clases, preferentemente cuadradas.
4. Preprocesar las imágenes.
5. Subir los datos obtenidos al Drive.
6. Hacer la predicción.

Predicción

```
# Hacer predicciones
df = pd.read_csv("/content/drive/MyDrive/Taller_IA/datos_2.csv")
test = df.values[:,1]
test = test.reshape((1,32,32))/255.0
print(test)
prediction = model.predict(test)
print(prediction)
print(prediction.argmax(axis=1))
```

```
[[[0.5372549  0.54509804 0.54509804 ... 0.60392157 0.60392157 0.60392157]
  [0.5372549  0.54117647 0.54509804 ... 0.6        0.6        0.60392157]
  [0.5372549  0.54117647 0.54901961 ... 0.6        0.60392157 0.60392157]
  ...
  [0.53333333 0.54117647 0.54509804 ... 0.58431373 0.58823529 0.58823529]
  [0.53333333 0.5372549  0.54509804 ... 0.58431373 0.58431373 0.58431373]
  [0.53333333 0.5372549  0.54117647 ... 0.58431373 0.58431373 0.58431373]]]
1/1 [==============================] - 0s 195ms/step
[[0.22876774 0.00820826 0.19814475 0.10966023 0.13706283 0.02308405
  0.02426015 0.00757392 0.25406325 0.00917487]]
[8]
```

Referencias

Aprendizaje profundo

https://www.geeksforgeeks.org/python-ai/

Práctica

https://www.analyticsvidhya.com/blog/2019/01/build-image-classification-model-10-minutes/

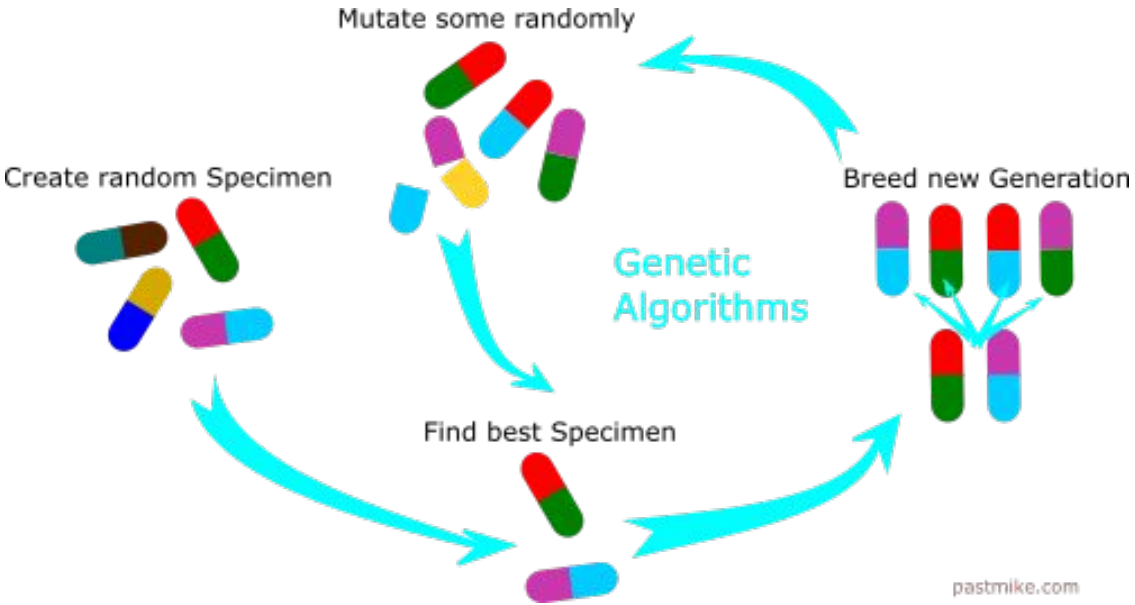
Data set

https://www.analyticsvidhya.com/blog/2022/01/image-classification-using-machine-learning/

Algoritmos Genéticos

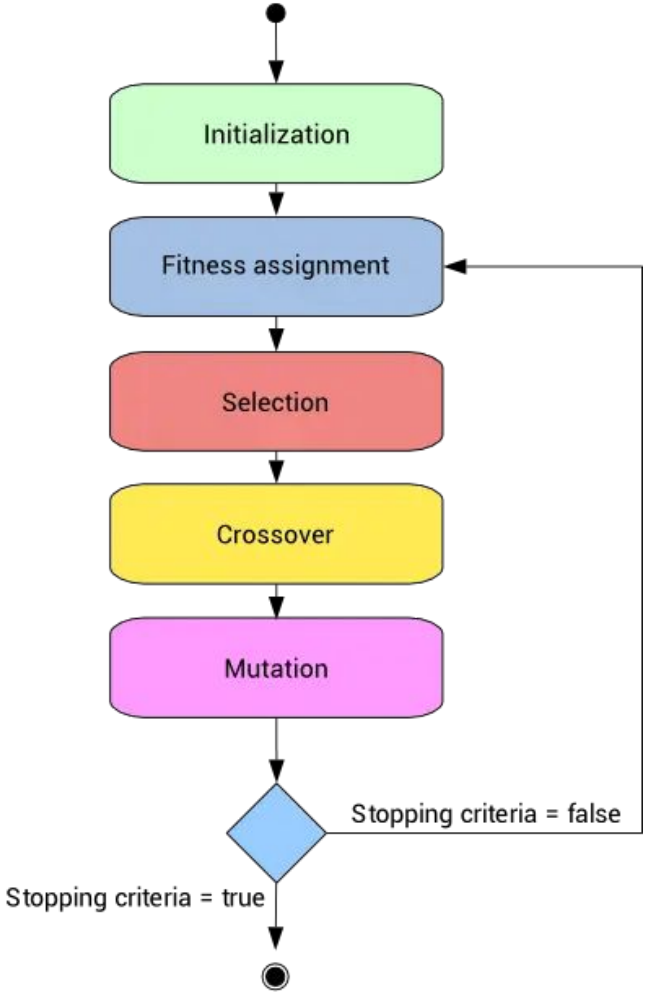
Algoritmos Genéticos

Su funcionamiento está inspirado en la teoría de la Evolución de las especies de Darwin, específicamente en la herencia genética y la selección natural.



Algoritmos Genéticos

Son capaces de generar soluciones de buena calidad para problemas de optimización y de búsqueda.



Práctica

Problema de optimización

Encontrar los parámetros $(x_1, x_2, x_3, x_4, x_5, x_6)$ que maximizan la función y

$$y = 4x_1 - 2x_2 + 3.5x_3 + 5x_4 - 11x_5 - 4.7x_6$$

donde $x_n \in (-4, 4)$

Abrir los archivos

… /Taller_IA/GA/Algoritmo_Genetico.py

… /Taller_IA/GA/ga.py

Parámetros de algoritmo genético

Población

Se le llama población a un conjunto de individuos. Cada individuo representa una solución al problema. En los algoritmos genéticos un individuo se representa como un vector de binarios, enteros, flotantes o letras.

Por ejemplo, un individuo i para la ecuación y tiene la forma:

$$y = 4x_1 - 2x_2 + 3.5x_3 + 5x_4 - 11x_5 - 4.7x_6$$

$$i = (x_1, x_2, x_3, x_4, x_5, x_6) = (3.2, 0.5, -1.7, -2.6, 1.2, -3.9)$$

Función de aptitud

Es el criterio que utilizaremos para determinar cuáles individuos son los mejor adaptados y seleccionarlos para apareamiento. Es muy importante porque es la guía del proceso de evolución.

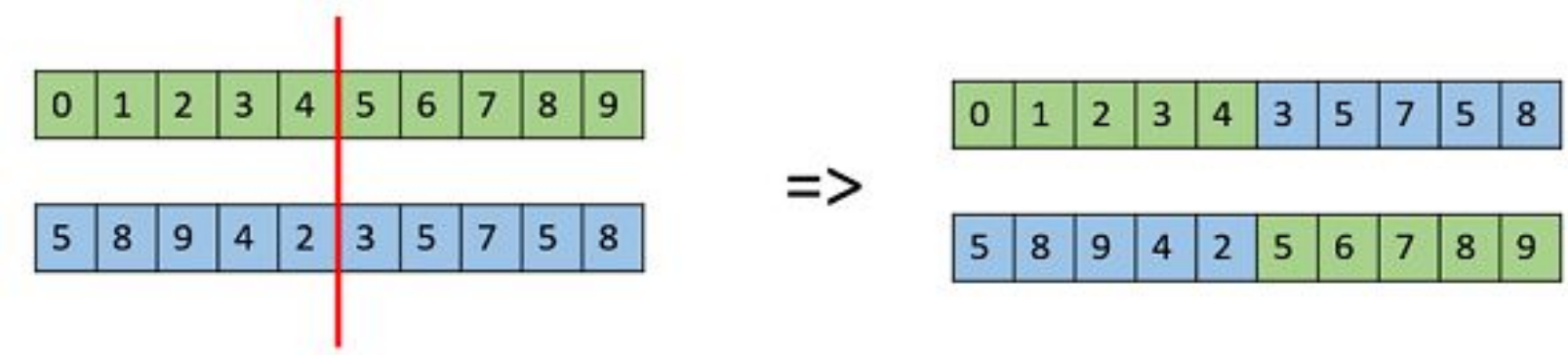
Por ejemplo: El valor f de aptitud de un individuo i es igual al valor de y obtenido cuando se calcula sustituyendo los valores de dicho individuo.

$$f_i = y_i = 4\ (3.2) - 2\ (0.5) + 3.5\ (-1.7) + 5\ (-2.6) - 11\ (1.2) - 4.7\ (-3.9)$$

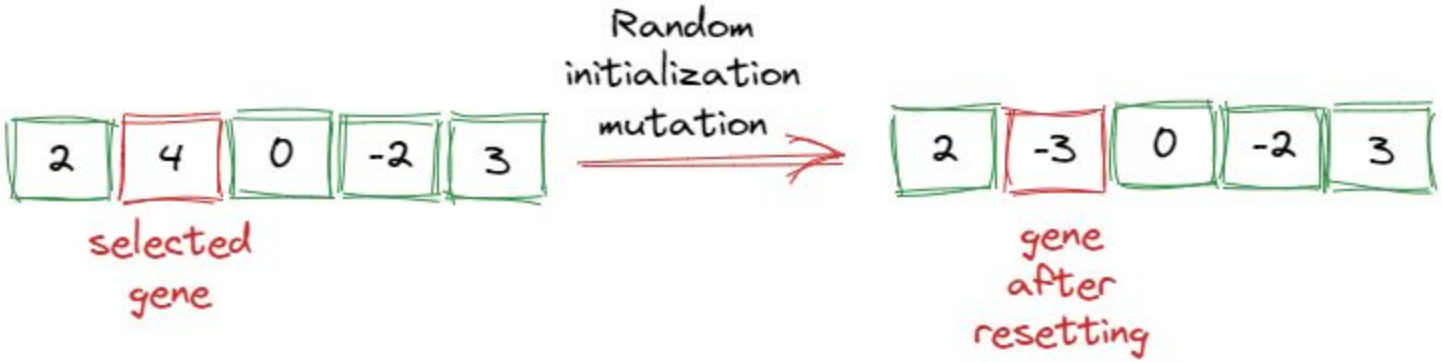
$$f_i = 12.8 - 1 - 5.95 - 13 - 13.2 + 18.33$$

$$f_i = -2.02$$

Cruzamiento

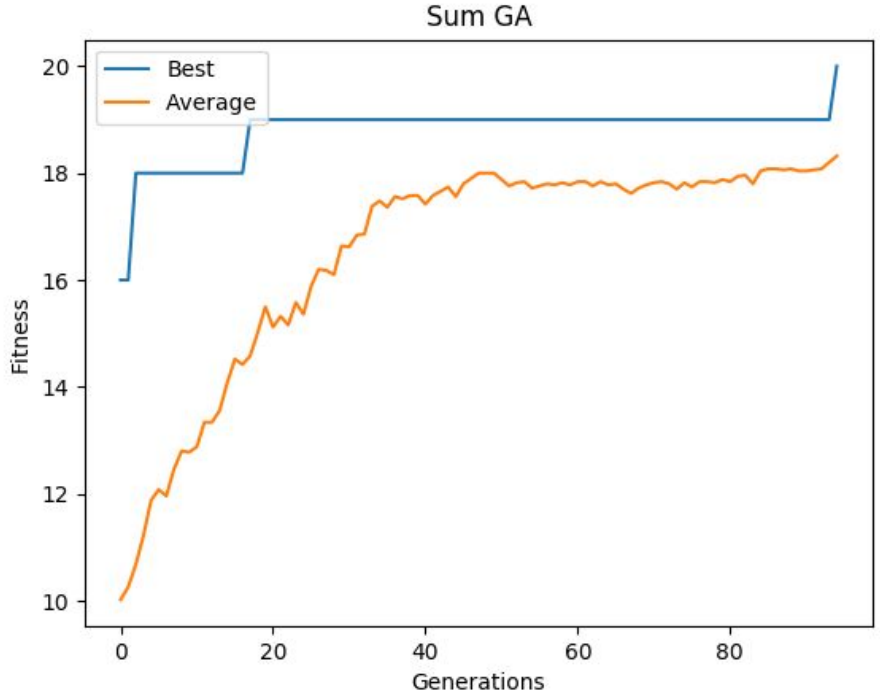


Mutación


2
4
0
-2
3
selected gene
Random initialization mutation
2
-3
0
-2
3
gene after resetting

Análisis de resultados

Graficar el fitness promedio y el mejor fitness de cada generación.



Referencias

Práctica

https://towardsdatascience.com/genetic-algorithm-implementation-in-python-5ab67bb124a6