Apuntes de Inteligencia Artificial clase: 02/16

Luis Molina
Profesor: Msc. Steven Andrey Pacheco Portuguéz

1 Discucion de la tarea

- **Yann LeCun:**

  Se considera uno de los padres de las redes neuronales convolucionales (CNN), las cuales revolucionaron el reconocimiento de imágenes.

- **Yoshua Bengio:** "Bengio ha hecho contribuciones clave en modelos probabilísticos de secuencias, utilizados para el reconocimiento de voz y de escritura y en aprendizaje no supervisado" [1]. También se considera el padre de las redes neuronales

- **Geoffrey Hinton:** Considerado también un padre de las redes neuronales. Hinton es el autor del algoritmo de retropropagación.

- **Timnit Gebru:** Ha realizado investigaciones acerca de los sesgos de los cuales sufren las AI y de los posibles dilemas éticos que surgen

- **Ian Goodfellow:** Inventor de las redes generativas adversarias (GAN). es cual es un avance crucial en el aprendizaje automático no supervisado.

- **Sam Altman:** Se considera uno de los padres de las redes neuronales convolucionales (CNN), las cuales revolucionaron el reconocimiento de imágenes.

1.1 Redes reuronales convolucionales

se distinguen de otras redes neuronales por su rendimiento superior con entradas de imagen, voz o señales de audio se utilizan con mayor frecuencia para tareas de clasificación y computer vision [2]

consisten de 3 capas:

- Covolucional: La capa convolucional es el bloque de creación principal de una CNN, y es donde se realizan la mayoría de los cálculos

- Agrupación: Permite reducir la dimensión mediante la reducción del número de parámetros de la entrada

- Totalmente conectada: Es la ultima capa de la red neuronal y esta completamente conectada cada nodo de esta capa con la capa anterior. Despues de aplicarle una función a esta capa es que se obtiene el "output" en función de probabilidad

1.2 Redes generativas adversarias

Este tipo de red consiste enrealidad de dos modelos que trabajan al mismo tiempo. Para su mejor entendimiento se pueden describir ambos modelos de la siguiente manera:

"El modelo generativo puede ser considerado análogo a un equipo de falsificadores, tratando de producir moneda falsa y usarla sin ser detectados, mientras que el modelo discriminativo es análogo a la policía, tratando de detectar la moneda falsificada. La competencia en este juego impulsa a ambos equipos a mejorar sus métodos hasta que las falsificaciones sean indistinguibles de los artículos genuinos." [3]

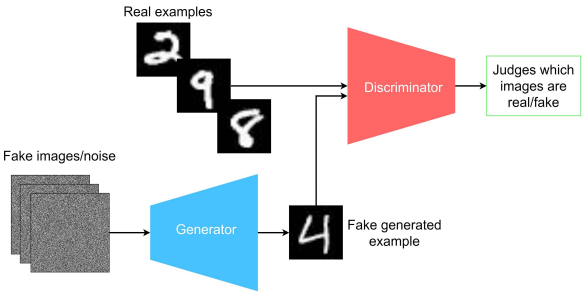

Figure 1: Visualización de los modelos [3]

1.3 Sora

Es el lanzamiento mas reciente de OpenAI, el cual es capaz de generar video de hasta 1 minuto de duracion dado un texto como su "input". los videos son de muy alta calidad y poseen una coherencia muy dislumbrante, sin embargo, toavia no esta disponible para el uso al publico [4]

1.4 Geminis

El modelo de inteligencia artificial de Google, Bard, se convirtio recientemente en Gemini [5] el cual posee nuevas y mas poderosas capacidades [6]

2 Introducción a Numpy

"Es una biblioteca fundamental en el ecosistema de Python para la programación científica y computacional. Proporciona estructuras de datos eficientes y funciones para trabajar con arreglos multidimensionales y realizar operaciones matemáticas y numéricas de manera eficiente." [7]

2.1 Usos

- **Arrays Multidimensionales:** Estos arrays permiten representar y manejar con mas eficiencia los vectores, matrices y tensores.
- **Operaciones Vectorizadas:** simplifica las operaciones entre vectores permitiendo operaciones de vectores, matrices o tensores enteros, en vez de uno por uno
- **Herramientas Estadísticas y matemáticas:** Variedad de funciones matemáticas y estadísticas integradas que facilitan el analisis posterior. incluyen: la descomposición de valores singulares, inversión de matrices y resolución de sistemas de ecuaciones lineales...
- **Integración con Otras Bibliotecas:** Es la base para otras bibliotecas en Python. ejemplos: SciPy, Pandas y Matplotlib.

2.2 Instalación

ver guia de instalación

2.3 Diferentes Creaciones de arreglos

```python
array1 = np.empty(2, int) #>> [213453, 756678247]
#arreglo con valores no inicializados

array2 = np.ones(5, int) #>> [1,1,1,1,1]
#arreglo con valores iniciales de 1

array3 = np.zeros(3, int) #>> [0,0,0]
#arreglo con valores iniciales de 0

array4 = np.empty_like(array3, int) #>> [2453, 756727, 87012]
#arreglo con la forma del array elegido pero inicializado segun funcion

array5 = np.ones_like(array1, int) #>> [1,1]
#arreglo con la forma del array elegido pero inicializado segun funcion

array6 = np.zeros_like(array2, int) #>> [0,0,0,0,0]
#arreglo con la forma del array elegido pero inicializado segun funcion

```

mas ejemplos en link

2.4 Creación de Matrices

```python
array1 = np.empty((2,1), int)
# [[1121],[43245]]

array2 = np.ones((1,3), int)
#[[1,1,1]]

array3 = np.zeros((3,2,2), int)
"""[[[0,0],
  [0,0]],

 [[0,0],
  [0,0]],

 [[0,0],
  [0,0]]]"""

array7 = np.identity(5, int) # esta es siempre n x n
"""[[1, 0, 0, 0, 0],
    [0, 1, 0, 0, 0],
    [0, 0, 1, 0, 0],
    [0, 0, 0, 1, 0],
    [0, 0, 0, 0, 1]]"""
```

mas ejemplos en link

2.5 Operación por dimensiones

```
matriz_2d = np.identity(5, int)
np.sum(matriz_2d) # suma todos los elementos de la matriz/tensor/arreglo
np.sum(matriz_2d, axis=1) # suma todos los elementos de la matriz/tensor/arreglo en la dimesion 1
np.sum(matriz_2d, axis=(1,4)) # suma todos los elementos de la matriz/tensor/arreglo en la dimesion 1 y 4

np.prod # equivalente pero multiplicando en vez de sumar
# ejemplo con matriz_2d: np.prod(matriz_2d, axis=0) == 1*0*0*0*0

```

mas ejemplos en link

2.6 Funciones estadíticas

```
array_random = np.random.rand(123)
media = np.mean(array_random)
desviacion_estandar = np.std(array_random)
valor_maximo = np.max(array_random)
valor_minimo = np.min(array_random)
```

mas ejemplos en link

2.7 Documentación

Documentación de Numpy

3 Introducción a Pandas

"Es una biblioteca que proporciona estructuras de datos y herramientas de análisis de datos" [7]

3.1 Usos

- **Creacion de data:** Permite la creación de volumenes significativos de información segun sean los paramentros requeridos.
- **Manipulacion de data:** Simplifica la manipulación de los datos de forma que se facilita su limpieza y analisis. permite operaciones como equivalentes a: "Select", "Filter", Agreggate, joining, merging...
- **I/O de data:** Permite la importación y exportación de los datos a diferentes formatos de diferentes formatos: CSV, Excel, SQL databases, JSON, HTML...
- **Visualización:** similar a los "Selects" de SQL permite visualizar los datos de una manera intuitiva y visual

3.2 Instalación

ver guia de instalación

3.3 Creación de Datasets

```
def createDataset(samples, seed = 42, plot=True, isTraining = True):
    # Establecer semilla para reproducibilidad
    np.random.seed(seed)
    #Creo clases de plantas con diferente distribuciones
    clase_1 = np.random.normal(loc=2, scale=1, size=(samples//3, 2))
    clase_2 = np.random.normal(loc=7, scale=1, size=(samples//3, 2))
    clase_3 = np.random.normal(loc=12, scale=1, size=(samples//3, 2))
    #aqui lo que sucede es que usando una distribucion normal se llenan valores random en un arreglo
    #para: clase_1, clase_2 y clase_3 que tendran "sample/3" elementos cada una


    # Etiquetas de clase
    etiquetas_clase_1 = np.full(samples//3, 0)  # 0 para clase 1
    etiquetas_clase_2 = np.full(samples//3, 1)  # 1 para clase 2
    etiquetas_clase_3 = np.full(samples//3, 2)  # 2 para clase 3
    #aqui simplemente estamos generando 3 arreglos de la forma [0,0...,0] , [1,1...,1] y [2,2...,2]

    # Combinar las clases en un solo dataset
    dataset = np.vstack((clase_1, clase_2, clase_3))
    # aqui se unifican los 3 arreglos con valores random
    # para explicar, a cada clases se les asignara una de las letras A, B y C:
    # la union resulta de la forma [A1,A2...,B1,B2...,C1,C2...]
    # pero esto es asi debido al ejemplo en concreto.


    etiquetas = np.concatenate((etiquetas_clase_1, etiquetas_clase_2, etiquetas_clase_3))
    #la funcion concatena los valores en la dimension especificada (default es 0)
```

```
    # en este caso es equivalente a [a1,aq2...,b1,b2...,c1,c2...]
    # sin embargo la funcion permite una concatenacion mas compleja si fuera el caso.
    #por lo que se recomienda un entendimiento mas profundo por parte del lector

    if plot:
        # Visualización de las clases en un scatter plot
        plt.scatter(clase_1[:, 0], clase_1[:, 1], label='Clase 1', alpha=0.7)
        plt.scatter(clase_2[:, 0], clase_2[:, 1], label='Clase 2', alpha=0.7)
        plt.scatter(clase_3[:, 0], clase_3[:, 1], label='Clase 3', alpha=0.7)
        plt.title('Dataset de Tres Clases con Distribuciones Diferentes')
        plt.xlabel('Longitud del sépalo')
        plt.ylabel('Longitud del pétalo')
        plt.legend()
        plt.show()
        # utilizando "matplotlib" se grafican los datos

    columnas = ['Longitud_sepalo', 'Longitud_petalo']
    df = pd.DataFrame(dataset, columns=columnas)
    #finalmente se crea un dataframe de pandas

    if isTraining:
        df['Clase'] = etiquetas
    # Barajar el DataFrame
    df = df.sample(frac=1).reset_index(drop=True
    return df

dataFrameTraining = createDataset(5000)
# se corre la función
```

3.4 Funciones de visualización de datos

```
# se da a entender que todo lo siguiente es de uso dentro de "print()"
dataFrameTraining.head(5)
dataFrameTraining.tail(5)
dataFrameTraining.count()
dataFrameTraining["Clase"] # valores de la columna especificada
dataFrameTraining[["Clase"]] # dataframe de la columna especificada
dataFrameTraining[["Longitud_petalo", "Clase"]] # varias columnas
dataFrameTraining.describe() # informacion estadistica de los datos
```

3.5 Funciones de visualización gráfica

las funciones a continuación no son de "Pandas". Le pertenecen a "Matplotlib"

```
import matplotlib as plt

# se da a conocer el unicamente el nombre de algunas de las funciones
plt.plot() # gráficos de líneas
plt.scatter() #gráficos de dispersión
plt.hist() # histogramas
plt.bar() # gráficos de barras
```

3.6 Documentació

Documentación de Pandas

4 K-Neareast Neighbor

Es un algoritmo del tipo "machine leaning". Este es no supervisado y muy simple de implementar. Pues consiste en la agrupacion de elementos segun un numero predefinido de agrupaciones de manera que la distancia entre cada punto y el centro de la agrupación sea reducia al minimo posible

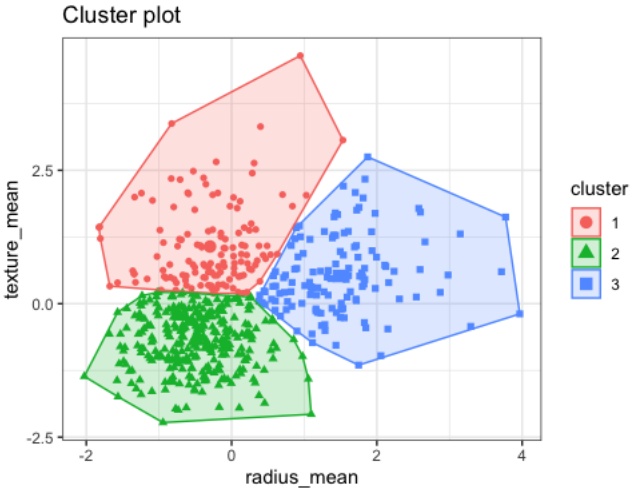

Figure 2: ejemplo grafico del algoritmo [8]

Debido a como fundamentalmente funciona el algoritmo, este no depende de un modelo para realizar predicciones, sino, que utiliza los mismo datos para realizar la predicción. lo cual originalmente suena como un beneficio pues el tiempo de entrenamiento es O(1) sin embargo, esta ganacia de tiempo se paga en el gasto de memoria el cual es O(ND) siendo N la cantidad de los datos y D las dimensionalidades de los datos

References

[1] F. princesa de Asturias. Geoffrey hinton, yann lecun, yoshua bengio y demis hassabis premio princesa de asturias de investigaciÓn cientÍfica y tÉcnica 2022. [Online]. Available: https://www.fpa.es/es/premios-princesa-de-asturias/premiados/2022-geoffrey-hinton-yann-lecun-yoshua-bengio-y-demis-hassabis.html?texto=trayectoria&especifica=0#:~:text=LeCun%20es%20autor%20o%20coautor,Nacional%20de%20Ingenier%C3%ADa%20de%20EE.

[2] IBM. ¿qué son las redes neuronales convolucionales? [Online]. Available: https://www.ibm.com/es-es/topics/convolutional-neural-networks

[3] C. Hansen. Generative adversarial networks explained. [Online]. Available: https://developer.ibm.com/articles/generative-adversarial-networks-explained/

[4] OpenAI. Creating video from text. [Online]. Available: https://openai.com/sora

[5] S. Hsiao. Bard becomes gemini: Try ultra 1.0 and a new mobile app today. [Online]. Available: https://blog.google/products/gemini/bard-gemini-advanced-app/

[6] S. Pichai. Introducing gemini: our largest and most capable ai model. [Online]. Available: https://blog.google/technology/ai/google-gemini-ai/#sundar-note

[7] S. Pacheco. Introducción a la programación vectorial. [Online]. Available: https://tecdigital.tec.ac.cr/dotlrn/classes/CA/IC6200/S-1-2024.CA.IC6200.2/file-storage/view/notebooks%2Fsupervised%2F3-Introducci%C3%B3n_Numpy.ipynb

[8] B. statistics collaboration of australia. K-means clustering. [Online]. Available: https://bookdown.org/andrew_grant/Unsupervised-learning/k-means-clustering.html