

Apuntes Semana 3 - 6/3/2025

Elaborado por: Tomás Granados Preciado

Abstract—Estos apuntes corresponden a la transcripción completa de la clase de Inteligencia Artificial del Grupo 2, Clase 4, Semana 3. Se abordan temas como álgebra lineal aplicada a IA, distancias euclidianas y de Manhattan, vectores, matrices, y el algoritmo de K-Vecinos Más Cercanos (K-Nearest Neighbors, KNN). Además, se incluyen ejemplos prácticos y explicaciones detalladas de los conceptos.

I. INTRODUCCIÓN

En esta clase, se revisaron conceptos fundamentales de álgebra lineal aplicados a la inteligencia artificial, como vectores, matrices, distancias euclidianas y de Manhattan. También se introdujo el algoritmo de K-Vecinos Más Cercanos (KNN) y se discutió su implementación en Python utilizando bibliotecas como NumPy y Pandas.

II. SITIOS RECOMENDADOS POR EL PROFESOR EN LA CLASE:

- Repositorio con papers actualizados diariamente relacionados con el mundo de AI: <https://github.com/dair-ai>
- Sitio con papers que traen código ejecutable junto a estos: <https://paperswithcode.com>

III. NOTICIAS DISCUTIDAS AL INICIO DE LA CLASE

A. Lanzamiento de GPT-4.5 para Usuarios Premium

Se habló sobre la liberación de GPT-4.5, una versión mejorada del modelo de lenguaje de OpenAI, disponible para usuarios Premium. Este modelo tiene mejoras en la capacidad de generar respuestas más coherentes y precisas.

B. Cadenas de Pensamiento (Chain-of-Thought)

Se discutió un concepto llamado **Chain-of-Thought** (Cadena de Pensamiento), que es una técnica utilizada en modelos de lenguaje como GPT. Esta técnica permite que el modelo "piense" paso a paso antes de generar una respuesta, lo que mejora su capacidad para resolver problemas complejos.

C. Noticias sobre NVIDIA y CUDA

Se mencionó un avance relacionado con **NVIDIA** y su tecnología **CUDA**, que permite acelerar el procesamiento de operaciones matemáticas en GPUs. Esto es especialmente útil para tareas de IA que requieren cálculos intensivos, como el entrenamiento de modelos.

IV. CONCEPTOS CLAVE

A. Vectores y Matrices

- Un **vector** es un arreglo n-dimensional que tiene dirección y magnitud.
- Los vectores se componen de un punto de origen y un punto final.
- La **magnitud** de un vector se calcula utilizando la distancia euclidiana.
- Las **matrices** son arreglos bidimensionales que permiten realizar operaciones lineales.

B. Distancias

- **Distancia Manhattan (L1)**: Se calcula sumando las diferencias absolutas entre las coordenadas de dos puntos.
- **Distancia Euclidiana (L2)**: Se calcula utilizando el teorema de Pitágoras para obtener la distancia más corta entre dos puntos.
- La distancia euclidiana es más precisa para medir distancias en espacios multidimensionales.

C. Producto Punto

- El **producto punto** entre dos vectores es una operación que devuelve un escalar.
- Se calcula como la sumatoria del producto de las entradas correspondientes de los vectores.
- El producto punto también puede interpretarse como la proyección de un vector sobre otro.

V. IMPLEMENTACIÓN PRÁCTICA

A. Programación Vectorial con NumPy

- NumPy es una biblioteca de Python que permite realizar operaciones vectoriales y matriciales de manera eficiente.
- Las operaciones vectorizadas en NumPy son más rápidas que los bucles tradicionales.
- Ejemplo de creación de matrices y vectores en NumPy:


```
import numpy as np
vector = np.array([1, 2, 3])
matriz = np.zeros((3, 3))
```

B. Algoritmo K-Vecinos Más Cercanos (KNN)

- El algoritmo KNN es un método de aprendizaje supervisado basado en instancias.
- Para clasificar una nueva instancia, se calculan las distancias entre esta y todas las instancias del conjunto de entrenamiento.
- Se seleccionan los K vecinos más cercanos y se realiza una votación para determinar la clase de la nueva instancia.

- Ejemplo de implementación en Python:

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)
predictions = knn.predict(X_test)
```

VI. EJEMPLOS Y EJERCICIOS

A. Ejemplo: Clasificación de Flores

- Se utilizó un dataset ficticio de flores con dos características: longitud del sépalo y longitud del pétalo.
- Se aplicó el algoritmo KNN para clasificar nuevas instancias basadas en las características del dataset.
- Se visualizaron los resultados utilizando gráficos de dispersión.

B. Ejercicio Propuesto

- Dado un conjunto de datos de producción sin etiquetar, clasificar cada instancia utilizando el algoritmo KNN.
- Calcular las distancias euclidianas entre las nuevas instancias y el conjunto de entrenamiento.
- Seleccionar los K vecinos más cercanos y determinar la clase de cada instancia.

VII. CONCLUSIÓN

En esta clase, se revisaron conceptos fundamentales de álgebra lineal y su aplicación en inteligencia artificial. Se introdujo el algoritmo KNN y se discutió su implementación en Python. Además, se realizaron ejercicios prácticos para afianzar los conocimientos.