

T.P Funciones comunes

1. Explicar el funcionamiento del código.
 2. Sacar a relucir los rasgos asociados a los distintos paradigmas de programación en el código.
 3. Explicar la función de cada script y cómo interactúan entre sí.
 4. Indicar los criterios y decisiones tomadas para el planteamiento del código.
 5. Explicar las fortalezas y debilidades del software diseñado
-

1. Explicación del Funcionamiento del Código

El código tiene como objetivo modelar y analizar datos utilizando distribuciones normales y funciones relacionadas con el análisis de señales de audio. Se divide en varias secciones:

Modelo de Tres Campanas:

tres_campanas: Combina tres distribuciones normales ("campanas") para generar un modelo compuesto. La función verifica la validez de los parámetros de entrada y calcula la contribución máxima de cada distribución en un rango de valores (x).

Cálculo del Error:

error: Calcula la diferencia entre dos conjuntos de datos usando una norma personalizada (parámetro **orden**). Esto es útil para medir la precisión del modelo.

Optimización con QPSO:

QPSO_tres_campanas: Implementa un algoritmo de optimización basado en Partículas Cuánticas para ajustar los parámetros del modelo de tres campanas a un conjunto de datos objetivo (X, Y).

Utiliza un enfoque iterativo para minimizar el error y refinar los parámetros en cada paso.

Filtrado de Señales:

filtrar_senal: Aplica filtros pasa bajos y pasa altos a una señal de entrada, utilizando la transformada de Butterworth para eliminar ruido y resaltar frecuencias relevantes.

Clases de Audio:

Audio: Proporciona herramientas para analizar señales de audio, incluyendo:

Limpieza y normalización en el dominio de la frecuencia.

Cálculo de frecuencias base, máximas y promedio mediante transformadas de Fourier.

Grabación de Audio:

Grabar: Captura una señal de audio mediante un micrófono, almacenando los datos en un archivo WAV.

2. Rasgos Asociados a Paradigmas de Programación

Programación Orientada a Objetos (POO):

La clase **Audio** encapsula funcionalidad para analizar señales de audio, favoreciendo la reutilización del código.

Programación Procedimental:

Las funciones como **Grabar**, **tres_campanas** y **error** siguen un flujo secuencial claro, diseñado para tareas específicas.

Programación Funcional:

Uso de funciones puras, como **norm.pdf**, y estructuras matemáticas claras en **tres_campanas**.

3. Explicación de los Scripts y su Interacción

Scripts y Componentes:

Modelo Matemático (tres_campanas**):**

Modela un conjunto de datos usando distribuciones normales escaladas.

Análisis de Audio (Audio**):**

Procesa señales de audio para extraer información clave como frecuencias dominantes.

Optimización (QPSO_tres_campanas):

Ajusta parámetros del modelo de tres campanas para minimizar errores respecto a datos observados.

Grabación y Filtrado:

Permiten capturar y procesar señales en tiempo real, asegurando que los datos de entrada sean de alta calidad.

Interacción:

El flujo típico podría ser:

Captura de una señal de audio.

Filtrado para eliminar ruido.

Análisis y cálculo de características clave (como la frecuencia base).

Ajuste de un modelo matemático (tres campanas) para representar los datos procesados.

Criterios y Decisiones del Planteamiento del Código

Uso de Algoritmos Estadísticos:

El modelo de tres campanas aprovecha las propiedades de distribuciones normales para describir datos complejos de manera compacta.

Optimización Cuántica (QPSO):

Elegida por su capacidad de explorar el espacio de soluciones eficientemente, incluso con restricciones complejas.

Enfoque Modular:

División clara entre captura de datos, procesamiento, modelado y optimización.

Criterios de Filtrado:

Los parámetros del filtro pasan altos y bajos aseguran que solo las frecuencias relevantes sean procesadas.

5. Fortalezas y Debilidades del Software Diseñado

Fortalezas:

Flexibilidad del Modelo:

El modelo de tres campanas puede adaptarse a una variedad de distribuciones y datos.

Robustez:

El filtro de señales y la limpieza en el dominio de frecuencia aseguran resultados confiables incluso con ruido moderado.

Optimizable:

QPSO permite refinar los parámetros del modelo, mejorando la precisión en cada iteración.

Reutilización:

La clase `Audio` y las funciones de filtrado pueden integrarse en otros proyectos de análisis de señales.

Debilidades:**Complejidad Computacional:**

QPSO puede ser costoso en términos de tiempo de cálculo, especialmente con un gran número de partículas y pasos.

Dependencia de Parámetros:

El rendimiento del modelo depende en gran medida de la elección de los parámetros iniciales y los valores del filtro.

Limitación del Modelo:

El modelo de tres campanas no es adecuado para datos que no se ajusten bien a distribuciones normales.