

Matplotlib

Origen

Ee remonta a un intento de uno de nosotros (John Hunter) de liberarse a sí mismo y a sus colegas investigadores de la epilepsia de un paquete de software patentado para realizar análisis de electrocorticografía (ECoG). El laboratorio en el que trabajaba solo tenía una licencia para el software, y los diversos estudiantes de posgrado, estudiantes de medicina, posdoctorados, pasantes e investigadores se turnaron para compartir el dongle de la llave de hardware

Rama dematplotlib se desarrolló originalmente como una herramienta de visualización de EEG/ECoG para esta aplicación GTK+, y este caso de uso dirigió su arquitectura original.

matplotlib se diseñó originalmente para cumplir un segundo propósito también: como un reemplazo para la generación de gráficos interactivos controlados por comandos, algo que MATLAB hace muy bien.

Qué es?

Es una biblioteca de trazado basada en Python con soporte completo para 2D y soporte limitado para gráficos 3D, ampliamente utilizada en la comunidad informática científica de Python. La biblioteca se dirige a una amplia gama de casos de uso. Puede incrustar gráficos en el kit de herramientas de interfaz de usuario de su elección y actualmente admite gráficos interactivos en todos los principales sistemas operativos de escritorio utilizando los kits de herramientas GTK+, Qt, Tk, FLTK, wxWidgets y Cocoa.

Descripción general de la arquitectura matplotlib

Una de las principales tareas arquitectónicas que matplotlib debe resolver es implementar un marco para representar y manipular lo Figureque está separado del acto de renderizarlo Figureen una ventana de interfaz de usuario o en una copia impresa.

La arquitectura para lograr esto se separa lógicamente en tres capas, que se pueden ver como una pila. Cada capa que se encuentra sobre otra capa sabe cómo hablar con la capa debajo de ella, pero la capa inferior no es consciente de las capas superiores. Las tres capas de abajo hacia arriba son: backend, artista y secuencias de comandos.

CAPA DE FONDO

En la parte inferior de la pila se encuentra la capa de back -end , que proporciona implementaciones concretas de las clases de interfaz abstractas: "FigureCanvas: encapsula el concepto de una superficie para dibujar (por ejemplo, "el papel"). "Renderer: hace el dibujo (por ejemplo, "el pincel"). "Event: maneja las entradas del usuario, como los eventos del teclado y el mouse.

CAPA DE ARTISTA

La Artistjeraquía es la capa intermedia de la pila de matplotlib y es el lugar donde ocurre gran parte del trabajo pesado. Siguiendo con la analogía de que el FigureCanvasdel backend es el papel, el Artistes el objeto que sabe tomar el Renderer(el pincel) y poner tinta en el lienzo. Todo lo que ves en un matplotlib Figurees una Artistinstancia; el título, las líneas, las etiquetas de verificación, las imágenes, etc., corresponden a Artistinstancias individuales

CAPA DE SECUENCIA DE COMANDOS

La secuencia de comandos que utiliza la API anterior funciona muy bien, especialmente para los programadores, y suele ser el paradigma de programación apropiado cuando se escribe un servidor de aplicaciones web, una aplicación de interfaz de usuario o quizás una secuencia de comandos para compartir con otros desarrolladores. Para propósitos cotidianos, particularmente para el trabajo exploratorio interactivo de científicos de banco que no son programadores profesionales, es un poco pesado sintácticamente.

Refactorización de back-end

dibujar_arco, dibujar_imagen, dibujar_linea_coleccion, dibujar_linea, dibujar_lineas, dibujar_punto, draw_quad_mesh, draw_polygon_collection, draw_polygon, draw_rectangle, dibujar_regpoly_coleccion

La lista completa de métodos API de back-end opcionales es: draw_markers: Dibuja un conjunto de marcadores. draw_path_collection: Dibuja una colección de caminos. draw_quad_mesh: Dibuja una malla cuadrilátera.

Transforma

matplotlib pasa mucho tiempo transformando coordenadas de un sistema a otro.

Estos sistemas de coordenadas incluyen: datos: los valores originales de datos sin procesar ejes: el espacio definido por un rectángulo de ejes particular figura: el espacio que contiene la figura completa visualización: las coordenadas físicas utilizadas en la salida (por ejemplo, puntos en PostScript, píxeles en PNG)

El oleoducto polilínea

Transformación: Las coordenadas se transforman de coordenadas de datos a coordenadas de figuras. Si se trata de una transformación puramente afin, como se describe anteriormente, es tan simple como una multiplicación de matrices. Si esto implica transformaciones arbitrarias, se llaman funciones de transformación para transformar las coordenadas en espacio figurativo.

Manejar datos faltantes: la matriz de datos puede tener partes en las que faltan datos o no son válidos. El usuario puede indicar esto configurando esos valores en NaNo usando numpymatrices enmascaradas. Los formatos de salida de vectores, como PDF, y las bibliotecas de representación, como Agg.

Recorte: los puntos fuera de los límites de la figura pueden aumentar el tamaño del archivo al incluir muchos puntos invisibles. Más importante aún, los valores de coordenadas muy grandes o muy pequeños pueden causar errores de desbordamiento en la representación del archivo de salida, lo que da como resultado una salida completamente distorsionada. Este paso de la canalización recorta la polilínea a medida que entra y sale de los bordes de la figura para evitar ambos problemas.

Ajuste: las líneas perfectamente verticales y horizontales pueden verse borrosas debido al antialiasing cuando sus centros no están alineados con el centro de un pixel (consulte la Figura 11.7). El paso de ajuste de la canalización primero determina si toda la polilínea se compone de segmentos horizontales y verticales (como un rectángulo alineado con el eje) y, de ser así, redondea cada vértice resultante al centro de pixel más cercano. Este paso solo se usa para backends ráster, ya que los backends vectoriales deben seguir teniendo puntos de datos exactos. Algunos procesadores de formatos de archivos vectoriales, como Adobe Acrobat, realizan un ajuste de píxeles cuando se ven en la pantalla.

Simplificación: al trazar gráficos muy densos, es posible que muchos de los puntos de la línea no sean visibles. Esto es particularmente cierto en los gráficos que representan una forma de onda ruidosa. La inclusión de estos puntos en el gráfico aumenta el tamaño del archivo e incluso puede alcanzar límites en la cantidad de puntos permitidos en el formato de archivo. Por lo tanto, se eliminan todos los puntos que se encuentran exactamente en la línea entre sus dos puntos vecinos

Texto matemático

Quizás la sintaxis más utilizada para las expresiones matemáticas es la del sistema de composición tipográfica TeX de Donald Knuth. Es una forma de convertir la entrada en un lenguaje de texto sin formato como este: \sqrt{\frac{\delta x}{\delta y}}

Pruebas de regresión

primer esfuerzo, se escribió un script que generó una serie de gráficos que ejercían varias funciones de matplotlib, en particular aquellas que eran difíciles de ejecutar correctamente.

segundo paso, se automatizó este enfoque general. El script de prueba actual de matplotlib genera una serie de gráficos, pero en lugar de requerir una intervención manual, esos gráficos se comparan automáticamente con las imágenes de referencia.

Lecciones aprendidas

Una de las lecciones importantes del desarrollo de matplotlib es, como dijo Le Corbusier, "los buenos arquitectos toman prestado". Los primeros autores de matplotlib eran en gran parte científicos, programadores autodidactas que intentaban hacer su trabajo, no científicos informáticos formados formalmente. Por lo tanto, no obtuvimos el diseño interno correcto en el primer intento.