Embedding Matplotlib Animations in Jupyter Notebooks

Louis Tiao — 2016-04-16 21:10 — 11 Comments — Source

UPDATE (March 2018): For a more modern solution that uses an interactive JavaScript widget, please go to my new post: Embedding Matplotlib Animations in Jupyter as Interactive JavaScript Widgets.

In his blog post Embedding Matplotlib Animations in IPython Notebooks, Jake VanderPlas presents a slick hack for embedding Matplotlib Animations in IPython Notebooks, which involves writing it as a video to a tempfile, and then re-encoding it in Base64 as a HTML5 Video.

Unfortunately (or rather fortunately), this hack has been largely rendered obsolete by the heavy development efforts dedicated to both Matplotlib and IPython Notebook (since renamed to Jupyter Notebook) in recent years. In particular, Matplotlib 1.5.1 now supports inline display of animations in the notebook with the `to_html5_video` method, which converts the animation to an h264 encoded video and embeddeds it directly in the notebook.

In this notebook, we reproduce Jake VanderPlas' blog post with this new feature.

In [1]:
```
%matplotlib inline
```

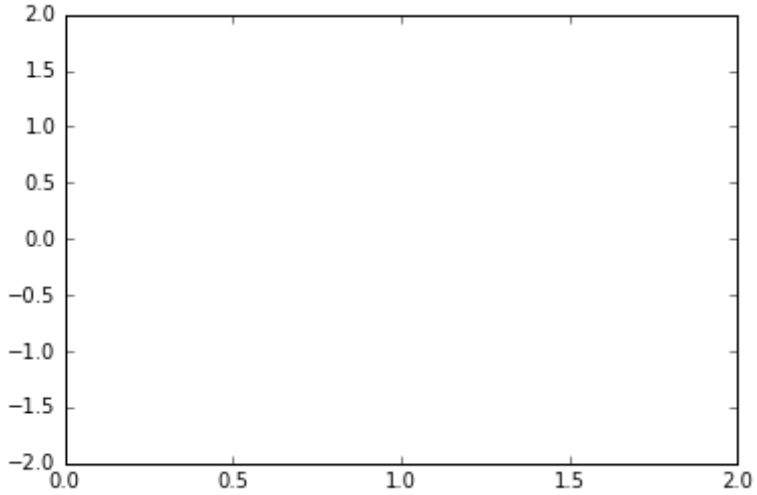
In [2]:
```
import numpy as np
import matplotlib.pyplot as plt

from matplotlib import animation, rc
from IPython.display import HTML
```

In [3]:
```
# First set up the figure, the axis, and the plot element we wa
fig, ax = plt.subplots()

ax.set_xlim(( 0, 2))
ax.set_ylim((-2, 2))

line, = ax.plot([], [], lw=2)
```



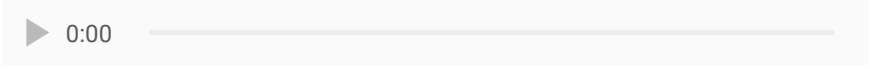
In [4]:
```
# initialization function: plot the background of each frame
def init():
    line.set_data([], [])
    return (line,)
```

In [5]:
```
# animation function. This is called sequentially
def animate(i):
    x = np.linspace(0, 2, 1000)
    y = np.sin(2 * np.pi * (x - 0.01 * i))
    line.set_data(x, y)
    return (line,)
```

In [6]:
```
# call the animator. blit=True means only re-draw the parts tha
anim = animation.FuncAnimation(fig, animate, init_func=init,
                               frames=100, interval=20, blit=Tr
```

In [7]:
```
HTML(anim.to_html5_video())
```

Out[7]:



Note that Animation instances now have a `_repr_html_` method. However, it returns `None` by default.

In [8]:
```
anim._repr_html_() is None
```

Out[8]:
```
True
```

This means we won't get any sort of animation from the inline display.

In [9]:
```
anim
```

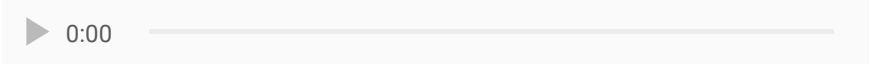
Out[9]:
```
<matplotlib.animation.FuncAnimation at 0x109421828>
```

The method used to display is controlled by the `animation.html` rc parameter, which currently supports values of `none` and `html5`. `none` is the default, performing no display. We simply need to set it to `html5`:

In [10]:
```
# equivalent to rcParams['animation.html'] = 'html5'
rc('animation', html='html5')
```

In [11]:
```
anim
```

Out[11]:



And that's all there is to it!

PREVIOUS POST NEXT POST

Comments

Your code works well for me, Thank you! But I fail as I try to animate a simpler function like y = 0.6 * x. I just want to see the line drawn across the window. Can you help me modify your code for that purpose? I have been trying to modify the lines def animate(i): and y = np.sin(2 * np.pi * (x - 0.01 * i)).

Reply • Share ›

Almost Nearly • 2 years ago

Very convenient. If you save the result of the .to_html5_video() call, is there an easy way to save it to a file? I want to render it in-notebook for preview, then once i'm happy, save it externally without having to re-render.

Reply • Share ›

helikotrema Almost Nearly • a year ago

You can try this:

```
anim.save('myvideo.mp4', codec='h264')
```

1 Reply • Share ›

Buzz Lightyear Almost Nearly • 2 years ago

I don't know if you was able to solve this. In my browser I get a download button so that I can "download" the video to mp4 file. But I would be interested in a command to save the video programmatically.

Reply • Share ›

Stevey B • 2 years ago

This is great! Thanks!

BUT...

How do I animate a contour plot, e.g. created using matplotlib.pyplot.pcolormesh?

Reply • Share ›

Aslam Abbas • 2 years ago

I got a player. But couldn't play it, as though nothing was rendered. I tried the same code as above. It looks exactly like the blank player that I can see in this post.
I also set matplotlib.rcParams['animation.writer'] = 'avconv'

[Update]
Failed in Chrome.
Successful in Firefox

 • Reply • Share ›

francois brest • 3 years ago

Hi,

Thank you for this post. Your method seems much better than Jake VanderPlas' one.
Unfortunately, I have an error:

RuntimeError Traceback (most recent call last)
<ipython-input-4-5114ccf53b4c> in <module>()
----> 1 HTML(anim.to_html5_video())

c:\winpython-64bit-3.5.1.2\python-3.5.1.amd64\lib\site-packages\matplotlib\animation.py in to_html5_video(self)
949 # We create a writer manually so that we can get the
950 # appropriate size for the tag
--> 951 Writer = writers[rcParams['animation.writer']]
952 writer = Writer(codec='h264',
953 bitrate=rcParams['animation.bitrate'],

c:\winpython-64bit-3.5.1.2\python-3.5.1.amd64\lib\site-packages\matplotlib\animation.py in __getitem__(self, name)

see more

 • Reply • Share ›

Louis Tiao Mod ↗ francois brest • 3 years ago

It definitely seems like it's having trouble finding ffmpeg, even though from your traceback it appears to be installed. Sorry I'm pretty hopeless with Windows! :(

^ | v • Reply • Share ›

francois brest ↗ Louis Tiao • 3 years ago

OK It worked after I reboot my PC. Sorry to have disturbed you.

^ | v • Reply • Share ›

ALSO ON **LOUIS TIAO PERSONAL**

Walkthrough: Deploying a Flask app with Redis Queue (RQ) Workers and …

1 comment • 3 years ago

Surya Pandian — Flask app with rq is working fine.After introducing SQLAlchemy rq making request to db is failing.

Implementing Variational Autoencoders in Keras: Beyond the Quickstart Tutorial

13 comments • a year ago

Louis Tiao — Just call `decoder.predict` on your latent vectors!Refer to my last example where I use this to generate an n x n grid of images …

Save Matplotlib Animations as GIFs

5 comments • 3 years ago

Fred Callaway — Another concise and informative tutorial. Thanks Louis!Running the Anaconda distribution 3.5 on a 2015 …

Upgrading all outdated pip packages

2 comments • 4 years ago

Louis Tiao — Looks awesome!