Track 4

Author: Kulpin Svyatoslav Sergeevich

Here I presented a very simple library – TenVis for visualizing 4D tensors 20x20x20x20, link. That library was written on Python, using matplotlib, numpy and PyQt. First of all, I want to discuss the instruments that I used to develop. Python was chosen for the following reason: the small size of the tensor (20x20x20x20), so the performance of this language will be sufficient, and if the library is written in Python, it is very easy to use it together with frameworks such as PyTorch, or TensorFlow. Matplotlib was used for making plots, numpy for work with arrays, PyQt for interactive plotting.

TenVis has 4 approaches to visualize a tensor: 3d projection, 2d projection, animation 3d projections along the chosen axis, animation 2d projections along the chosen axis. Axes in TenVis are numbered, as in numpy, starting from 0. TenVis has a class Tensor, that class has methods for visualization. Next, we will look at class Tensor and each visualization method in detail with examples.

class Tensor(tensor, copy=True)

a class for visualizing 4D tensor 20x20x20x20

Parameters:

- tensor : np.darray
  The 4D tensor to visualize.
- copy : bool
  If True, a copy of the tensor is made. If False, the tensor is used as-is.

Attributes:

- tensor : np.darray
  The array for visualization.

Tensor.show_3d(axis, index, size=(24, 13.5), max_value=255, min_value=0, alpha=1, log=False, colorbar=True, file_name=None, show=True)

Visualize 3D slice of the tensor

Parameters:

- axis : int
  Axis for the slice. Should be {0, 1, 2, 3}.
- index : int
  The index by which the slice of the selected axis is assigned. Should be {0, …, 19}.
- size : (float, float)
  Size of the plot.
- max_value : int
  Maximum value for filtration.
- min_value : int
  Minimum value for filtration.
- alpha : float
  The alpha blending value, between 0 (transparent) and 1 (opaque).

- log : bool
  Use a log scale.
- colorbar : bool
  Use a colorbar.
- file_name : str
  The name to save the file. If None, the file will not be saved.
- show : bool
  Show a plot.

Example 1:

```python
import tenvis
import numpy as np

array = np.random.randint(0, 256, size=(20, 20, 20, 20), dtype=np.uint8)

array[1, :, 3:10, :] = 0
t = tenvis.Tensor(array, copy=False)

t.show_3d(0, 1)

array[1, :, :, 17: ] = 0
t.show_3d(0, 1)
```

Here I created a NumPy array. Then, I modified the array slightly (I will often change the array in the examples to demonstrate that the library correctly displays the projections). After that, I created a **Tensor()** object, with **copy=False**. Next I displayed a 3D projection using **show_3d**. In this projection, the index of the **axis0** is set 1. We can observe a large black region because the values with **axis2** coordinates between 3 and 9 are set to 0 (Fig. 1). Afterward, I modified the parent array again, and again displayed the same projection. Since **copy=False**, these changes are reflected in the projection (Fig. 2)

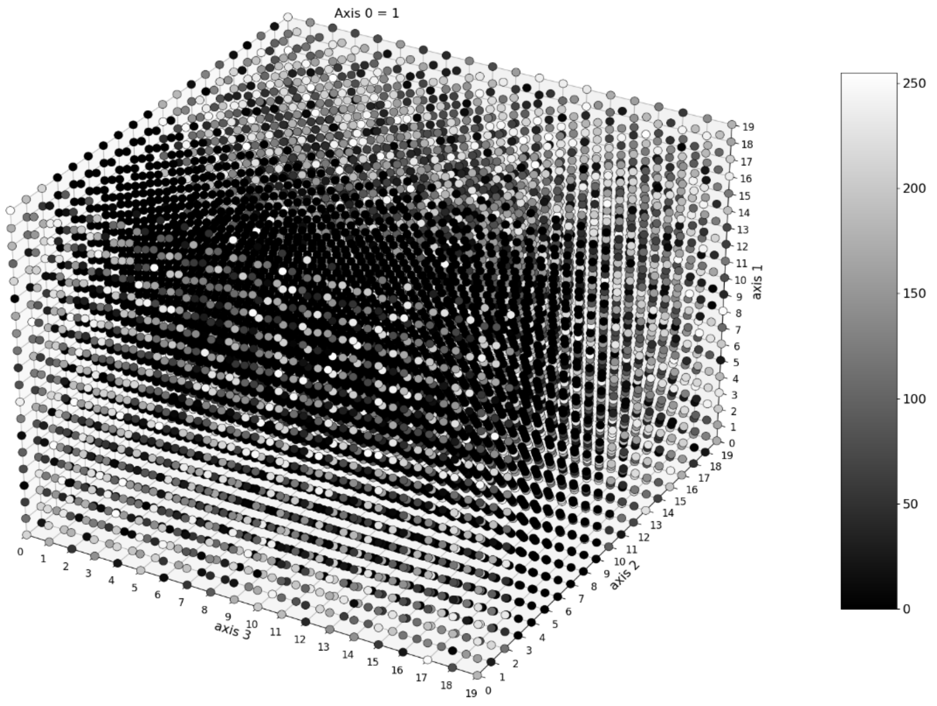

Fig. 1

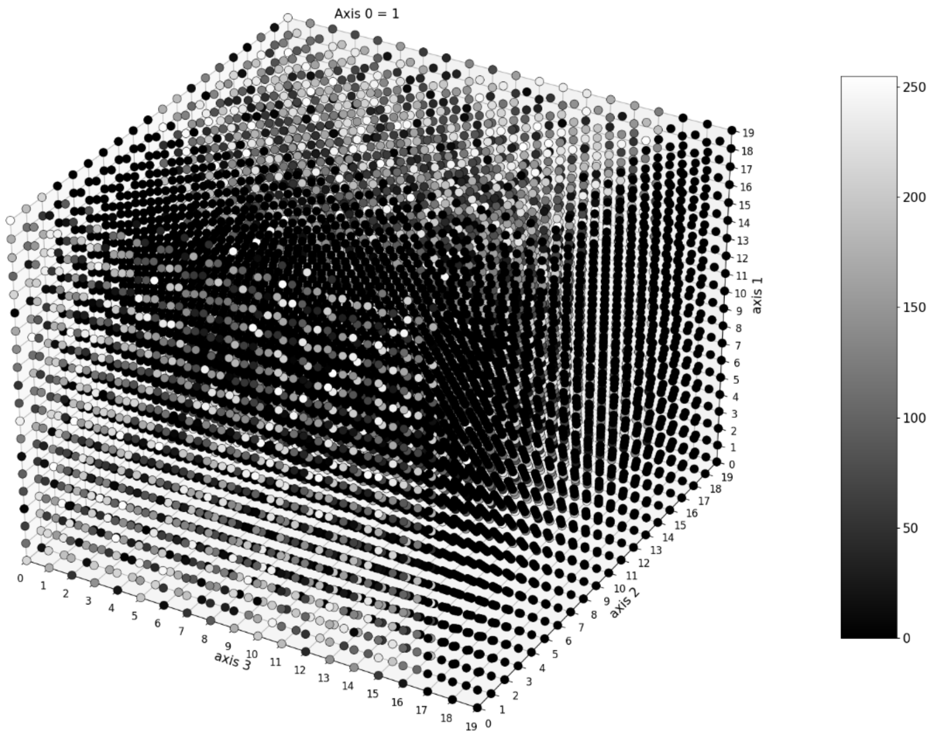

Fig.2

Example 2:

```python
import tenvis
import numpy as np

array = np.random.randint(0, 256, size=(20, 20, 20, 20), dtype=np.uint8)

```

```python
t = tenvis.Tensor(array)

t.show_3d(2, 7, log=True)
```

In this example, I also displayed 3D projection with log scale (Fig. 3). In this projection, the index of the **axis2** is set 7.

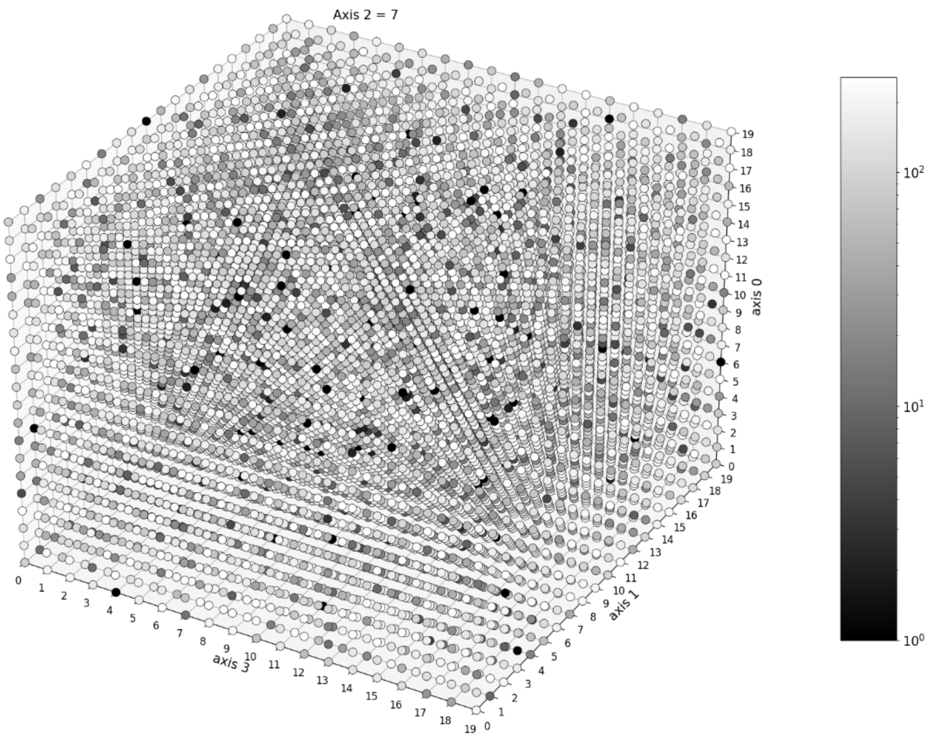

Fig. 3

Example 3:

```python
import tenvis
import numpy as np

array = np.random.randint(0, 256, size=(20, 20, 20, 20), dtype=np.uint8)

array[:, 5, 5:7, :] = 255
array[:, 5, :, 15:17] = 255

t = tenvis.Tensor(array)

t.show_3d(1, 5, min_value=230, alpha=0.9,
   file_name="example3_show3d.png")
```

I displayed 3D projection with filtration (Fig. 4). There are only points on the projection whose value is >= 230. I made the dots a little transparent and saved locally the plot. We can observe two white plates along the **axis2** and **axis3**.

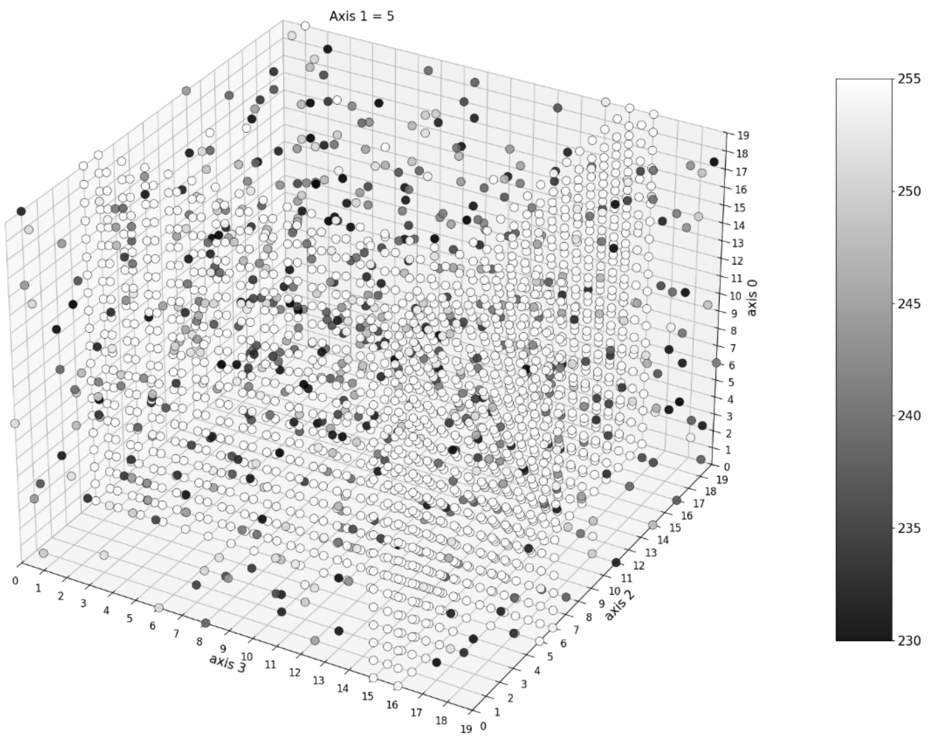

Fig. 4

Tensor.show_2d(axis1, index1, axis2, index2, size=(24, 13.5), log=False, colorbar=True, file_name=None, show=True)

Visualize 2D slice of the tensor

Parameters:

- axis1 : int
  First axis for the slice. Should be {0, 1, 2, 3}.
- index1 : int
  The index by which the slice of the selected first axis is assigned. Should be {0, ..., 19}.
- axis2 : int
  Second axis for the slice. Should be {0, 1, 2, 3}
- index2 : int
  The index by which the slice of the selected second axis is assigned. Should be {0, ..., 19}.
- size : (float, float)
  Size of the plot.
- log : bool
  Use a log scale.
- colorbar : bool
  Use a colorbar.
- file_name : str
  The name to save the file. If None, the file will not be saved.
- show : bool

Show a plot.

Example 4:

```python
import tenvis
import numpy as np

array = np.random.randint(0, 256, size=(20, 20, 20, 20), dtype=np.uint8)
array[7, 10, :, 15] = 25
t = tenvis.Tensor(array)

t.show_2d(1, 10, 3, 15, file_name="example4_show2d.png")

array2 = np.random.randint(0, 256, size=(20, 20, 20, 20), dtype=np.uint8)
array2[2, 5, 7, :] = 25
t = tenvis.Tensor(array2)

t.show_2d(0, 2, 1, 5, log=True)
```

I displayed two 2d slice. In first 2d projection the index of the **axis1** is set 10 and index of the **axis3** is set 15 (Fig. 5). And another 2d slice, with log scale (Fig.6)

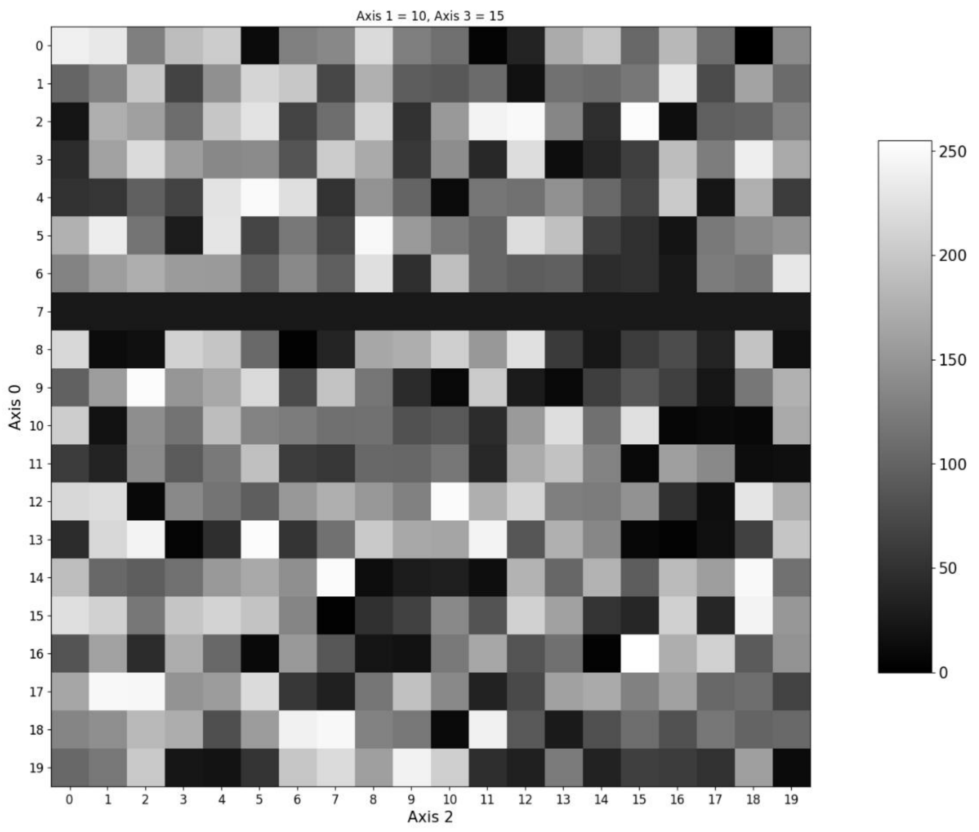

Fig. 5

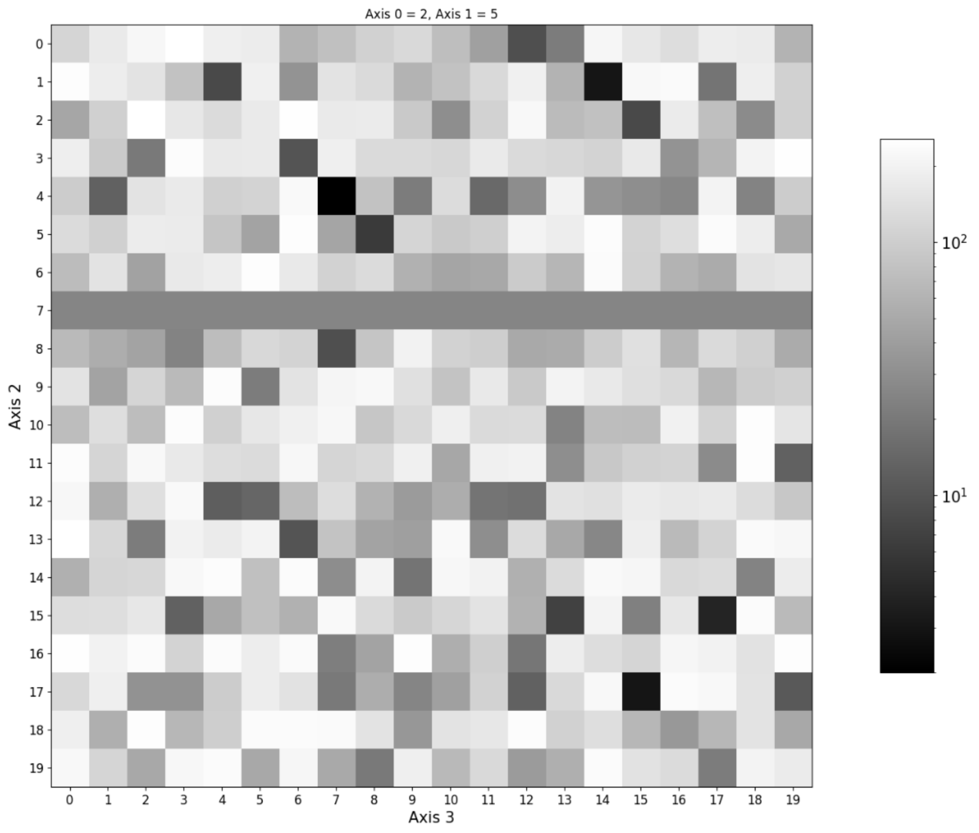

Fig. 6

Tensor.animate_3d(axis, size=(24, 13.5), max_value=255, min_value=0, alpha=1, log=False, colorbar=True, file_name=None, interval=1000, show=True)

Animation 3D slices along the chosen axis.

Parameters:

- axis : int
  The axis along which the animation will take place. Should be {0, 1, 2, 3}.
- size : (float, float)
  Size of the plot.
- max_value : int
  Maximum value for filtration.
- min_value : int
  Minimum value for filtration.
- alpha : float
  The alpha blending value, between 0 (transparent) and 1 (opaque).
- log : bool
  Use a log scale.
- colorbar : bool
  Use a colorbar.
- file_name : str
  The name to save the file. If None, the file will not be saved.
- interval : int
  Delay between frames in milliseconds.
- show : bool

Show a plot.

Example 5:

```python
import tenvis
import numpy as np

array = np.random.randint(0, 256, size=(20, 20, 20, 20), dtype=np.uint8)
for i in range(18):
    array[i, i + 2, :, :] = 0
    array[i, i + 1, :, :] = 0
    array[i, i, :, :] = 0
t = tenvis.Tensor(array)

t.animate_3d(0, colorbar=False, max_value=100, interval=2000, alpha=0.8,
   file_name="example5_animate3d.gif")
```

I displayed animation of the 3D projections (each frame is just a slice along the selected axis)(Fig. 7). (if the animation doesn't work, then look at the github repositories in the examples folder.)

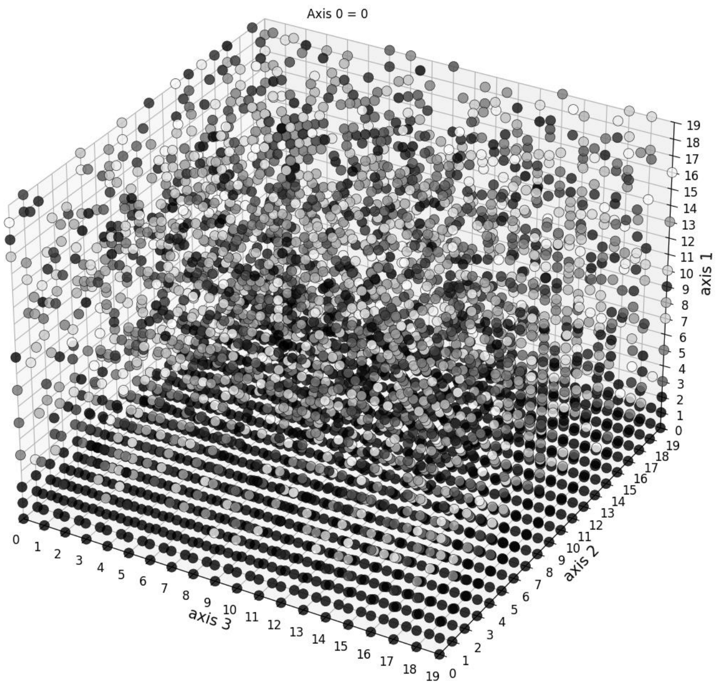

Fig. 7

Tensor.animate_2d(axis1, index1, axis2, size=(24, 13.5), log=False, colorbar=True, file_name=None, interval=1000, show=True)

Animation 2D slices along the chosen axis.

Parameters:

- axis1 : int
  Axis for the slice. Should be {0, 1, 2, 3}.
- index1 : int
  The index by which the slice of the selected axis is assigned. Should be {0, ..., 19}.
- axis2 : int

The axis along which the animation will take place. Should be {0, 1, 2, 3}

- size : (float, float)
  Size of the plot.
- log : bool
  Use a log scale.
- colorbar : bool
  Use a colorbar.
- file_name : str
  The name to save the file. If None, the file will not be saved.
- interval : int
  Delay between frames in milliseconds.
- show : bool
  Show a plot.

Example 6:

```python
import tenvis
import numpy as np

array = np.random.randint(0, 256, size=(20, 20, 20, 20), dtype=np.uint8)
for i in range(19):
    if i % 2 == 0:
        array[i, 5, 5, :] = 255
t = tenvis.Tensor(array)

t.animate_2d(2, 5, 0, file_name="example6_animate2d.gif")
```

Here I displayed animation of the 2D slices (Fig. 8) (if the animation doesn't work, then look at the github repositories in the examples folder.)

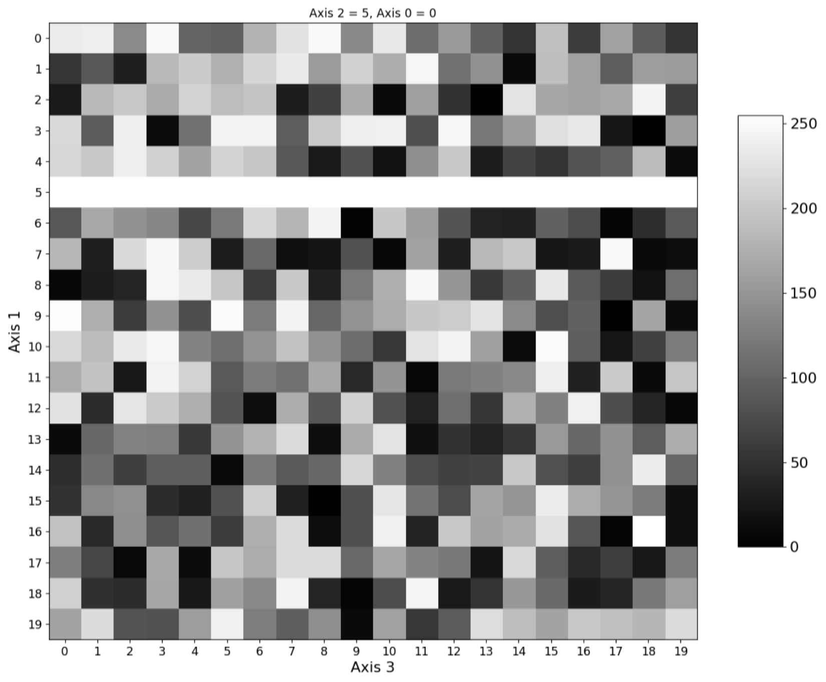

Fig. 8

In conclusion, several improvements could enhance the library’s functionality. These include displaying multipe projections simultaneously, providing tensor statistics, enabling tensor transformations, supporting PyTorch and TensorFlow tensors, and offering more flexible graph customization options (e.g., color selection, partial projections and point shapes). However, my goal was to create a very simple tensor visualization library. This task was a great learning experience and I enjoyed working on this project.